

Министерство образования Республики Беларусь

Министерство науки и высшего образования Российской Федерации

К.А. Волосов, Н.К. Волосова, А.К. Волосова,
Д.Ф. Пастухов, Ю.Ф. Пастухов

ПРАКТИКУМ

ДЛЯ ЗАДАЧ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА

Учебное пособие к лекционным и практическим занятиям

для студентов специальности

1-39 03 02 Искусственный интеллект

Московский институт инженеров транспорта

Полоцкий государственный университет имени Евфросинии Полоцкой

Москва, Новополоцк
2026

УДК 519.6

Рецензенты:

М.И. Карлов, кандидат физико-математических наук, защитил диссертацию на Механико-математическом факультете МГУ им. М.В. Ломоносова;

Волосов К.А., Волосова Н.К., Волосова А.К., Пастухов Д.Ф., Пастухов Ю.Ф.

Практикум для задач искусственного интеллекта: учебное пособие // К.А. Волосов, Н.К. Волосова, А.К. Волосова Д.Ф. Пастухов, Ю.Ф. Пастухов, - 1 - ое изд. - Москва: МИИТ, ПГУ. 2026. – 100 С.

Материал учебного пособия соответствует требованиям Государственного образовательного стандарта по математике и адресован студентам–математикам и студентам–программистам. Книга разбита на четыре основные главы, содержащие вспомогательные математические задачи с преобразованием точечных множеств, а также задачу оптимизации однослойной нейронной сети. Рассмотрены три модели однослойной нейронной сети. Функционал ошибок равен не сумме квадратов разности функции учителя и функции обученной сети, а сумме четвертых степеней указанных разностей. Предложенные программы одномерной сети независимы от промышленных нейронных сред, самостоятельны и просты в использовании.

В первое издание включены работы с одномерным, двухмерным и трехмерным дискретным преобразованием Фурье, приведены примеры. Дискретные преобразования Фурье используется во многих технических приложениях. Обобщено определение четной и нечетной функции для функции нескольких переменных, а также определение четной или нечетной функции для дискретной функции нескольких натуральных индексов.

Для студентов университетов, педагогических и технических вузов, преподавателей, инженеров, программистов, использующих в практической деятельности нейронные сети, элементы статистики, преобразования Фурье, математические преобразования точечных множеств на плоскости.

УДК 519.6

Российский Университет Транспорта, Полоцкий государственный университет имени Евфросинии Полоцкой, 2026.

Содержание

Введение	4
Глава 1. Лекция 1. Некоторые сведения из теории вероятностей.	6
Практическая работа 1. Одномерное гауссово распределение и его свойства.	7
Практическая работа 2. Двухмерное гауссово распределение и его свойства.	10
Практическая работа 3. Задача-навигатор на плоскости.	14
Глава 2. Математические преобразования над точечными множествами	18
Практическая работа 4. Слабый и сильный критерии отделимости двух точечных множеств на плоскости.	18
Практическая работа 5. Слабый и сильный критерии отделимости двух точечных множеств в пространстве.	21
Практическая работа 6. Задача поиска главных направлений точечного множества на плоскости	25
Глава 3. Несколько вариантов математического описания однослойной нейронной сети.	27
Лекция 2. Задача первая. Прямоугольная матрица плюс вектор весовых коэффициентов на входе нейронного монослоя. Практическая работа 7.	27
Лекция 3. Задача вторая. Прямоугольная матрица плюс вектор весовых коэффициентов на входе и квадратная матрица на выходе нейронного монослоя. Практическая работа 8.	35
Лекция 4. Задача третья. Прямоугольная матрица плюс вектор весовых коэффициентов на входе и квазиединичная квадратная матрица на выходе нейронного монослоя. Практическая работа 9.	45
Глава 4. Дискретное преобразование Фурье.	53
Лекция 5. Одномерное дискретное преобразование Фурье.	53
Практическая работа 10. Одномерное дискретное преобразование Фурье.	55
Лекция 6. Двухмерное и трехмерное дискретное преобразование Фурье.	59
Практическая работа 11. Двухмерное дискретное преобразование Фурье	60
Практическая работа 12. Трехмерное дискретное преобразование Фурье	66
Литература	74

Введение

Учебное пособие Практикум для задач искусственного интеллекта предназначен для студентов программистов и студентов математических специальностей. Курс рассчитан на один семестр и содержит 12 практических работ. Студенты вкладываются по времени в одно-два занятия для написания кода программы каждой практической работы.

Первая лекция и связанные с ней две практические работы посвящены свойствам равномерного и свойствам нормального распределения, одномерного и двухмерного. Принято считать, что в задачах по теории вероятностей, если результат имеет ошибку в 5% (одна значащая цифра результата) это уже хорошо. Однако в данных работах с нормальным распределением случайной величины мы получили численные результаты с 14-15 значащими цифрами, то есть, с двойной точностью на эффективном отрезке и эффективном прямоугольнике со сторонами равными 10-ти кратному среднеквадратичному отклонению сигма по обе стороны от среднего значения.

Вторая часть учебного пособия посвящена свойствам одного или двух точечных множеств. Задача-навигатор о поиске минимальной длины незамкнутого графа Эйлера при фиксированных положениях вершин на плоскости и задача поиска главных направлений точечного множества на плоскости (код программы с кратким описанием алгоритма). В этой части также расположены 2 работы с использованием слабого и сильного критерия отделимости двух точечных множеств на плоскости и в пространстве.

Третья часть содержит три лекции и три практических работы с оптимизацией однослойной нейронной сети. Входной сигнал умножается на весовую матрицу и складывается со свободным вектором и только затем покомпонентно преобразуется целевой функцией нейрона (модифицированной сигмоидой), всего $(n+1)m$ входных весовых параметров для оптимизации нейронной сетью. На выходе нейрона сигнал умножается на весовую квадратную матрицу размерности $m \cdot m$. Методом градиентного спуска в цикле в пространстве размерности $m(m+n+1)$ изменяются весовые коэффициенты. После цикла оптимизации принимается решение на каждом выходе (всего m выходов) однослойной нейронной сетью выбрать ближайшее значение из множества $\{-1,+1\}$. Такой же цифровой сигнал $\{-1,+1\}$ поступает на каждый вход нейронной сети $\{-1,+1\}$ (всего n входов). Для увеличения значения градиента функционал ошибок выбран не в виде суммы квадратов разности функции учителя и сигнала обученной нейронной сети для всех возможных входных состояний сигнала, а функционал 4-х степеней данных разностей. С помощью данной сети решается задача распознавания цвета предмета или задача сжатия цветности в целое число раз (максимальное цветов входных сигналов равно 16). Оказывается, что если происходит перестановка цветов или число сжатия равно 1,2,4,8,9,10,11,12,13,14,15,16, то есть, либо коэффициент сжатия цветовой гаммы нацело делит 16, либо он более 8, то обучение нейронной сети равно 100%.

В задаче распознавания, перестановки и сжатия гаммы цветов худший процент правильного обучения нейронной сети составил 81,26 %, а средневзвешенный процент правильного обучения по всем входным состояниям сигнала равен 96,1%. Академик РАН, профессор Московского государственного университета Анатолий Иванович Ларкин сформулировал основы современной теории одномерных электронных систем. Аналогично идеям Анатолия Ивановича в данной работе мы рассматриваем однослойную нейронную сеть и оптимизацию сети.

В четвертой части приведены лекции с выводом формул прямого и обратного дискретного преобразования Фурье, одномерного, двухмерного и трехмерного. Во-первых, такие преобразования очень часто используются в технических приложениях. А во-вторых, дискретные преобразования Фурье могут служить отдельным аппаратом для интерпретации результатов в задачах обработки нейронной сетью, например, одномерного звукового сигнала, плоского или пространственного сигнала.

Отметим также, что при решении задач нейронной сетью, возникают обратные задачи управления некоторой технической системой и требуют применения теории экстремальных задач и задач оптимального управления, например, с помощью принципа максимума Понтрягина [1], [2], [3]. В работах [90], [91], [92], [93] приведены результаты управления робототехнической системой, в которой процесс управления выбирает два крайних значения ускорения робота из множества параметра управления (ускорение робота).

Работы, связанные с нейронной сетью приведены в литературе под номерами [10] - [48]. Многие работы по вычислительной математике [4] - [9], [49] - [54], [68] - [89], [94] - [215] важны для использования эффективных вычислительных алгоритмов нейронной сетью. В работах [55] - [66], [212] - [282] нейронная сеть способна найти фрагменты на фотографиях с заданными свойствами или регистрировать быстро движущиеся объекты.

Лекция 1. Некоторые сведения из теории вероятностей.

В статистике и в теории вероятностей крайними или диаметрально противоположными являются два распределения – равномерное распределение и гауссово распределение функции плотности вероятностей.

Случайная величина x распределена на отрезке $[-a, a]$ равномерно, если ее плотность вероятностей равна постоянной величине $\rho(x) = \frac{1}{2a}, x \in [-a, a]$ (1.1)

Среднее значение равномерно распределенной величины x равно нулю

$$\bar{x} = \int_{-a}^a x \rho(x) dx = \frac{1}{2a} \int_{-a}^a x dx = \frac{x^2}{4a} \Big|_{-a}^a = 0$$

Дисперсия случайной величины (второй момент) определяется формулой

$$D = \int_{-a}^a (x - \bar{x})^2 \rho(x) dx = \frac{1}{2a} \int_{-a}^a x^2 dx = \frac{x^3}{6a} \Big|_{-a}^a = \frac{a^2}{3}, \sigma = \sqrt{D} = \frac{a}{\sqrt{3}}$$

То есть, среднеквадратическое отклонение равномерного распределения прямо пропорционально длине отрезка $2a$.

Нормальное распределение на прямой задается двумя параметрами (\bar{x}, σ) формулой

$$\rho(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x - \bar{x})^2}{2\sigma^2}\right), x \in [-\infty, \infty] \quad (1.2)$$

Смысл параметров (\bar{x}, σ) выясним, вычисляя первый и второй моменты для случайной величины, распределенной с учетом формулы (2.1).

$$\begin{aligned} \bar{x} &= \int_{-\infty}^{\infty} x \rho(x) dx = \frac{1}{\sqrt{2\pi}\sigma} \int_{-\infty}^{\infty} (x - \bar{x} + \bar{x}) \exp\left(-\frac{(x - \bar{x})^2}{2\sigma^2}\right) d\left(\frac{x - \bar{x}}{\sqrt{2\sigma}}\right) \stackrel{y = \frac{x - \bar{x}}{\sqrt{2\sigma}}}{=} \frac{\sqrt{2\sigma}}{\sqrt{\pi}} \int_{-\infty}^{\infty} \left(\frac{x - \bar{x} + \bar{x}}{\sqrt{2\sigma}}\right) \exp(-y^2) dy = \\ &= \frac{\sqrt{2\sigma}}{\sqrt{\pi}} \int_{-\infty}^{\infty} \left(y + \frac{\bar{x}}{\sqrt{2\sigma}}\right) \exp(-y^2) dy = \frac{\sqrt{2\sigma}}{\sqrt{\pi}} \int_{-\infty}^{\infty} y \exp(-y^2) dy + \frac{\bar{x}}{\sqrt{\pi}} \int_{-\infty}^{\infty} \exp(-y^2) dy = \\ &= 0 + \frac{\bar{x}}{\sqrt{\pi}} \int_{-\infty}^{\infty} \exp(-y^2) dy = \frac{\bar{x}}{\sqrt{\pi}} \sqrt{\pi} = \bar{x} \end{aligned}$$

Первый интеграл равен нулю, так как подынтегральная функция $y \exp(-y^2)$ является нечетной на симметричном интервале $(-\infty, +\infty)$. А второй интеграл вычислим через двойной интеграл, обозначая

$$\begin{aligned} I_0 &= \int_{-\infty}^{\infty} \exp(-y^2) dy. I_0^2 = \int_{-\infty}^{\infty} \exp(-y^2) dy \cdot \int_{-\infty}^{\infty} \exp(-x^2) dx = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \exp(-(x^2 + y^2)) dx dy = \\ &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \exp(-(x^2 + y^2)) dx dy = \int_0^{\infty} 2\pi \rho \exp(-\rho^2) d\rho = -\pi \int_0^{\infty} \exp(-\rho^2) d(-\rho^2) = \\ &= -\pi \exp(-\rho^2) \Big|_0^{\infty} = \pi(-0 + \exp(0)) = \pi = I_0^2 \Leftrightarrow I_0 = \sqrt{\pi} \end{aligned}$$

Таким образом, первый параметр \bar{x} равен первому моменту или среднему значению случайной величины, распределенной по нормальному закону. Вычислим второй момент случайной величины относительно точки \bar{x} на числовой прямой.

$$\begin{aligned}
 D &= \int_{-\infty}^{\infty} (x - \bar{x})^2 \rho(x) dx = \frac{1}{\sqrt{2\pi}\sigma} \int_{-\infty}^{\infty} (x - \bar{x})^2 \exp\left(-\frac{(x - \bar{x})^2}{2\sigma^2}\right) d(x - \bar{x}) \stackrel{y = \frac{x - \bar{x}}{\sqrt{2}\sigma}}{=} \frac{2\sigma^2}{\sqrt{\pi}} \int_{-\infty}^{\infty} \left(\frac{x - \bar{x}}{\sqrt{2}\sigma}\right)^2 \exp(-y^2) dy = \\
 &= \frac{2\sigma^2}{\sqrt{\pi}} \int_{-\infty}^{\infty} y^2 \exp(-y^2) dy = \frac{-\sigma^2}{\sqrt{\pi}} \int_{-\infty}^{\infty} y(\exp(-y^2))' dy = \frac{-\sigma^2}{\sqrt{\pi}} y \exp(-y^2) \Big|_{-\infty}^{+\infty} + \frac{\sigma^2}{\sqrt{\pi}} \int_{-\infty}^{\infty} \exp(-y^2) dy = \\
 &= \frac{\sigma^2}{\sqrt{\pi}} \int_{-\infty}^{\infty} \exp(-y^2) dy = \frac{\sigma^2}{\sqrt{\pi}} I_0 = \frac{\sigma^2}{\sqrt{\pi}} \sqrt{\pi} = \sigma^2 \Leftrightarrow \sigma = \sqrt{D}
 \end{aligned}$$

Замечание 1. В преобразованиях использовалась производная $(\exp(-y^2))' = -2y \exp(-y^2)$.

Замечание 2. Принято считать, что в задачах по теории вероятностей полученный результат с ошибкой 30% или с ошибкой 5% является надежным и хорошим (при этом результат имеет всего одну верную значащую цифру).

В следующих двух лабораторных работах для проверки свойств одномерного и двумерного нормальных распределений можно получать результаты с двойной точностью, однако для этого нужно использовать эффективную прямую расположенную на отрезке симметрично относительно среднего значения и шириной 10σ по обе стороны.

Практическая работа 1. Одномерное гауссово распределение и его свойства.

Нормальное распределение на прямой задается двумя параметрами (\bar{x}, σ) формулой

$$\rho(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x - \bar{x})^2}{2\sigma^2}\right), x \in [-\infty, \infty] \quad (1)$$

Введем отрезок “эффективной прямой” $x \in [\bar{x} - 10\sigma, \bar{x} + 10\sigma]$. Проверим три основных свойства нормального распределения на “эффективной прямой”:

- 1) $\int_{\bar{x} - 10\sigma}^{\bar{x} + 10\sigma} \rho(x) dx = 1$ - условие нормировки нормального распределения
- 2) $\int_{\bar{x} - 10\sigma}^{\bar{x} + 10\sigma} x \rho(x) dx = \bar{x}$ - среднее значение нормального распределения
- 3) $\int_{\bar{x} - 10\sigma}^{\bar{x} + 10\sigma} (x - \bar{x})^2 \rho(x) dx = \sigma^2$ - дисперсия нормального распределения

Мы использовали квадратурную формулу [245], [270], [272], [281] с двенадцатым порядком погрешности для вычисления определенного интеграла с помощью интегральной квадратурной формулы, при котором отрезок интегрирования делится на число интервалов кратное 10.

$$\langle y_1, y_2 \rangle = \int_a^b y_1(x) y_2(x) dx = 5h \sum_{i=0}^{n_1} y_1(x_i) y_2(x_i) C_i + O(h^{12}), n = 10p, h = \frac{b-a}{n}, p \in N \quad (2)$$

где весовые коэффициенты интегральной квадратурной формулы (2) определяются величиной остатка по модулю 10 от номера узла равномерной сетки i .

$$C_i = \begin{cases} \frac{16067}{299376}, & i = 0 \text{ или } i = n_1, \\ \frac{16067}{149688}, & (i \equiv 0 \pmod{10}) \text{ и } (0 < i < n_1), \\ \frac{26575}{74844}, & \text{если } (i \equiv 1 \pmod{10}) \text{ или } (i \equiv 9 \pmod{10}), \\ \frac{-16175}{99792}, & \text{если } (i \equiv 2 \pmod{10}) \text{ или } (i \equiv 8 \pmod{10}), \\ \frac{5675}{6237}, & \text{если } (i \equiv 3 \pmod{10}) \text{ или } (i \equiv 7 \pmod{10}), \\ \frac{-4825}{5544}, & \text{если } (i \equiv 4 \pmod{10}) \text{ или } (i \equiv 6 \pmod{10}), \\ \frac{17807}{12474}, & \text{если } i \equiv 5 \pmod{10}. \end{cases}$$

Напишем код программы на распространенном языке C++. Среднее значение, средне-квадратичное отклонение, правый конец и левый конец эффективной прямой равны соответственно $\sigma=2.0$; $s=7.0$; $a=s-10.0*\sigma$; $b=s+10.0*\sigma$.

```
#include "stdafx.h"
#include<stdio.h>
#include<math.h>
double f(double x,double sigma,double s)
{
    double s1,s2,pi;
    pi=2.0*asin(1.0);
    s1=((x-s)*(x-s))/(2.0*sigma*sigma);
    s2=sigma*sqrt(2.0*pi);
    return exp(-s1)/s2;
}
int const n=300;
int main()
{
    int i;
    double a,b,h,y,eps,s,sigma;
    double c[n+1],c0[8];
    double ss,x;
    c0[1]=16067.0/299376.0; c0[2]=26575.0/74844.0;
    c0[3]=-16175.0/99792.0; c0[4]=5675.0/6237.0;
    c0[5]=-4825.0/5544.0; c0[6]=17807.0/12474.0;
    c0[7]=16067/149688.0;
    sigma=2.0;s=0.0;a=s-10.0*sigma;b=s+10.0*sigma;
    h=(b-a)/double(n);
    for(i=0;i<=n;i++)
    {
        if(i==0 || i==n)
        {
            c[i]=c0[1];
        }
        if(i%10==1 || i%10==9)
        {
            c[i]=c0[2];
        }
    }
}
```

```

    }
    if(i%10==2 || i%10==8)
    {
c[i]=c0[3];
    }
    if(i%10==3 || i%10==7)
    {
c[i]=c0[4];
    }
    if(i%10==4 || i%10==6)
    {
c[i]=c0[5];
    }

    if(i%10==5)
    {
c[i]=c0[6];
    }

    if(i%10==0 && i>0 && i<n)
    {
c[i]=c0[7];
    }
}

```

```

ss=0.0;
for(i=0;i<=n;i++)
{
x=a+h*double(i);
ss=ss+h*c[i]*f(x,sigma,s);
}
ss=ss*5.0;
y= ss;
printf("y=%.16lf\n",y);
eps=y-1.0;
printf("eps=%.16lf\n",eps);
}

```

1)Программа показывает результат($f(x,\sigma,s)= \exp(-s1)/s2$)

y=1.0000000000000000
eps=0.0000000000000000

Press any key to continue

eps=y-1.0;

2)Программа показывает результат($f(x,\sigma,s)=x* \exp(-s1)/s2$)

y=7.0000000000000018
eps=0.0000000000000018

Press any key to continue

eps=y-s;

3)Программа показывает результат($f(x,\sigma,s)= (x-s)*(x-s)* \exp(-s1)/s2$)

y=4.0000000000000027
eps=0.0000000000000027

Press any key to continue

eps=y- sigma* sigma;

Практическая работа 2. Двухмерное гауссово распределение и его свойства.

Нормальное распределение на плоскости имеет четыре параметра $(\bar{x}_1, \bar{x}_2, \sigma_1, \sigma_2)$ и получается формальным перемножением двух независимых нормальных распределений по переменным x, y с параметрами (\bar{x}_1, σ_1) и (\bar{x}_2, σ_2)

$$\begin{aligned} \rho(x, y, \bar{x}, \bar{y}, \sigma_1, \sigma_2) &= \frac{1}{\sqrt{2\pi}\sigma_1\sqrt{2\pi}\sigma_2} \exp\left(-\frac{(x-\bar{x})^2}{2\sigma_1^2}\right) \exp\left(-\frac{(y-\bar{y})^2}{2\sigma_2^2}\right) = \\ &= \frac{1}{2\pi\sigma_1\sigma_2} \exp\left(-\frac{(x-\bar{x})^2}{2\sigma_1^2} - \frac{(y-\bar{y})^2}{2\sigma_2^2}\right), x, y \in [-\infty, \infty] \quad (1) \end{aligned}$$

Введем прямоугольник “эффективной плоскости” $x \in [\bar{x} - 10\sigma_1, \bar{x} + 10\sigma_1], y \in [\bar{y} - 10\sigma_2, \bar{y} + 10\sigma_2]$. Проверим три основных свойства нормального распределения на “эффективной плоскости”:

$$1) \int_{\bar{x}-10\sigma_1}^{\bar{x}+10\sigma_1} \int_{\bar{y}-10\sigma_2}^{\bar{y}+10\sigma_2} \rho(x, y) dx dy = 1 - \text{условие нормировки нормального распределения}$$

$$2) \int_{\bar{x}-10\sigma_1}^{\bar{x}+10\sigma_1} \int_{\bar{y}-10\sigma_2}^{\bar{y}+10\sigma_2} xy \rho(x, y) dx dy = \bar{x}\bar{y} - \text{среднее значение смешанного произведения}$$

переменных x, y для нормального распределения

$$3) \int_{\bar{x}-10\sigma_1}^{\bar{x}+10\sigma_1} \int_{\bar{y}-10\sigma_2}^{\bar{y}+10\sigma_2} (x-\bar{x})^2 (y-\bar{y})^2 \rho(x, y) dx dy = \sigma_1^2 \sigma_2^2 - \text{произведение дисперсий нормального}$$

распределения по координатам x и y .

Мы использовали квадратурную формулу [245], [270], [272], [281] с двенадцатым порядком погрешности для вычисления двойного определенного интеграла на прямоугольнике с помощью интегральной квадратурной формулы, при котором каждый из двух отрезков интегрирования делится на число интервалов кратное 10.

$$\begin{aligned} \langle g_1, g_2 \rangle &= \iint_{a,c}^{b,d} g_1(x, y) g_2(x, y) dx dy = 25h_1 h_2 \sum_{i=0}^{n_1} \sum_{j=0}^{n_2} C_{i,j} g_1(x_j, y_i) g_2(x_j, y_i) + \\ &+ O(h_1^{12} + h_2^{12}), n_1 = 10p_1, h_1 = \frac{b-a}{n_1}, n_2 = 10p_2, h_2 = \frac{d-c}{n_2}, p_1, p_2 \in N, C_{i,j} = C_i^1 C_j^2, i = \overline{0, n_1}, j = \overline{0, n_2} \quad (2) \end{aligned}$$

где весовые коэффициенты интегральной квадратурной формулы (3) определяются величиной остатка по модулю 10 от номера узла равномерной сетки i, j .

$$C_i^2 = \begin{cases} \frac{16067}{299376}, i = 0 \text{ или } i = n_2, \\ \frac{16067}{149688}, (i \equiv 0 \pmod{10}) \text{ и } (0 < i < n_2), \\ \frac{26575}{74844}, \text{если } (i \equiv 1 \pmod{10}) \text{ или } (i \equiv 9 \pmod{10}), \\ \frac{-16175}{99792}, \text{если } (i \equiv 2 \pmod{10}) \text{ или } (i \equiv 8 \pmod{10}), \\ \frac{5675}{6237}, \text{если } (i \equiv 3 \pmod{10}) \text{ или } (i \equiv 7 \pmod{10}), \\ \frac{-4825}{5544}, \text{если } (i \equiv 4 \pmod{10}) \text{ или } (i \equiv 6 \pmod{10}), \\ \frac{17807}{12474}, \text{если } i \equiv 5 \pmod{10}. \end{cases} \quad (3)$$

$$C_j^1 = \begin{cases} \frac{16067}{299376}, j = 0 \text{ или } j = n_1, \\ \frac{16067}{149688}, (j \equiv 0 \pmod{10}) \text{ и } (0 < j < n_1), \\ \frac{26575}{74844}, \text{если } (j \equiv 1 \pmod{10}) \text{ или } (j \equiv 9 \pmod{10}), \\ \frac{-16175}{99792}, \text{если } (j \equiv 2 \pmod{10}) \text{ или } (j \equiv 8 \pmod{10}), \\ \frac{5675}{6237}, \text{если } (j \equiv 3 \pmod{10}) \text{ или } (j \equiv 7 \pmod{10}), \\ \frac{-4825}{5544}, \text{если } (j \equiv 4 \pmod{10}) \text{ или } (j \equiv 6 \pmod{10}), \\ \frac{17807}{12474}, \text{если } j \equiv 5 \pmod{10}. \end{cases} \quad (4)$$

Напишем программу на языке C++. Средние значения, среднеквадратичные отклонения, правые концы и левые концы эффективной плоскости равны соответственно $\sigma_1=2.0; \sigma_2=1.0; \sigma_1=7.0; \sigma_2=5.0; a=\sigma_1-10.0*\sigma_1; b=\sigma_1+10.0*\sigma_1; c=\sigma_2-10.0*\sigma_2; d=\sigma_2+10.0*\sigma_2;$

Далее следует код программы с библиотекой для среды Visual Studio

```
#include "stdafx.h"
#include<stdio.h>
#include<math.h>
double f(double x,double y,double sigma1,double sigma2,double ss1,double ss2)
{
    double s1,s2,pi;
    pi=2.0*asin(1.0);
    s1=((x-ss1)*(x-ss1))/(2.0*sigma1*sigma1)+((y-ss2)*(y-ss2))/(2.0*sigma2*sigma2);
    s2=sigma1*sigma2*(2.0*pi);
    return exp(-s1)/s2;
}
int const n1=200,n2=200;
int main()
{
```

```

int i,j;
double a,b,c,d,h1,h2,y,eps,ss1,sigma1,ss2,sigma2;
double c2[n1+1][n2+1],c3[n1+1],c4[n2+1],c0[8];
double ss,x;
c0[1]=16067.0/299376.0; c0[2]=26575.0/74844.0;
c0[3]=-16175.0/99792.0; c0[4]=5675.0/6237.0;
c0[5]=-4825.0/5544.0; c0[6]=17807.0/12474.0;
c0[7]=16067/149688.0;
sigma1=2.0;ss1=3.0;sigma2=1.0;ss2=5.0;a=ss1-10.0*sigma1;b=ss1+10.0*sigma1;
c=ss2-10.0*sigma2;d=ss2+10.0*sigma2;h1=(b-a)/double(n1);h2=(d-c)/double(n2);
for(i=0;i<=n2;i++)
{
    if(i==0 || i==n2)
    {
        c4[i]=c0[1];
    }
    if(i%10==1 || i%10==9)
    {
        c4[i]=c0[2];
    }
    if(i%10==2 || i%10==8)
    {
        c4[i]=c0[3];
    }
    if(i%10==3 || i%10==7)
    {
        c4[i]=c0[4];
    }
    if(i%10==4 || i%10==6)
    {
        c4[i]=c0[5];
    }
    if(i%10==5)
    {
        c4[i]=c0[6];
    }
    if(i%10==0 && i>0 && i<n2)
    {
        c4[i]=c0[7];
    }
}
for(j=0;j<=n1;j++)
{
    if(j==0 || j==n1)
    {
        c3[j]=c0[1];
    }
    if(j%10==1 || j%10==9)
    {
        c3[j]=c0[2];
    }
    if(j%10==2 || j%10==8)
    {

```

```

c3[j]=c0[3];
}
    if(j%10==3 || j%10==7)
    {
        c3[j]=c0[4];
    }
    if(j%10==4 || j%10==6)
    {
        c3[j]=c0[5];
    }
    if(j%10==5)
    {
        c3[j]=c0[6];
    }
    if(j%10==0 && j>0 && j<n1)
    {
        c3[j]=c0[7];
    }
}

```

////////////////////////////////////

```

for(i=0;i<=n2;i++)
for(j=0;j<=n1;j++)
{
c2[i][j]=c3[j]*c4[i];
}
ss=0.0;
for(i=0;i<=n2;i++)
for(j=0;j<=n1;j++)
{
    x=a+h1*double(j);y=c+h2*double(i);
    ss=ss+c2[i][j]*f(x,y,sigma1,sigma2,ss1,ss2);
}
ss=ss*25.0*h1*h2;
y=ss;
printf("y=%.16lf\n",y);
eps=y-1.0;
printf("eps=%.16lf\n",eps);
}

```

1)Программа показывает результат($f(x, y, \sigma_1, \sigma_2, s_1, s_2) = x \cdot y \cdot \exp(-s_1)/s_2$)

y=1.0000000000000000

eps=0.0000000000000000

Press any key to continue

eps=y-1.0;

2)Программа показывает результат($f(x, y, \sigma_1, \sigma_2, s_1, s_2) = x \cdot y \cdot \exp(-s_1)/s_2$)

y=14.9999999999999360 $\approx 3 \cdot 5$

eps=-0.00000000000000639

Press any key to continue

eps=y-ss1*ss2; n1=200,n2=500;

3)Программа показывает результат($f(x, y, \sigma_1, \sigma_2, s_1, s_2) = (x-ss1)^* (x-ss1)*(y-ss2)^* (y-ss2)^* \exp(-s_1)/s_2$)

y=3.9999999999999352 $\approx 2^2 \cdot 1^2$

eps=-0.00000000000000648

Press any key to continue

eps=y- sigma1* sigma1* sigma2* sigma2; n1=300,n2=300;

Замечание. Двухмерное нормальное распределения на плоскости в данной работе нетрудно также обобщить на трехмерный случай (нормальное распределение в пространстве).

Практическая работа 3. Задача-навигатор на плоскости.

Рассмотрим задачу – навигатор об оптимальной траектории с минимальной суммарной длиной звеньев ребер, соединяющих все вершины – адресатов (пациентов), которых нужно обойти почтальону или врачу, чтобы сократить время своей работы.

Координаты всех вершин заданы, как и задано начальное положение почтальона (5.0;5.0). Методом перебора по девяти свободным вершинам (по девяти целочисленным переменным) построить Эйлеров граф, в котором при движении почтальона все адресаты – вершины встречаются ровно один раз.

Программа-навигатор для простого графа Эйлера с минимальной суммой длин ребер написана на языке C++, методом исключения – во внутренних циклах не могут встретиться индексы вершин, если они были задействованы уже во внешних циклах. Каждая оптимальная последовательность вершин на данный момент запоминается программой, пока ее не сменяет другая оптимальная последовательность вершин с еще меньшей суммарной длиной ребер. Длина ребер вычисляется по теореме Пифагора отдельной программой-функцией.

```
#include "stdafx.h"
#include<stdio.h>
#include<math.h>
double r(double x1,double y1, double x2,double y2)
{ return sqrt((x2-x1)*(x2-x1)+(y2-y1)*(y2-y1));
}
int const l=9;
int main()
{
    int i1,i2,i3,i4,i5,i6,i7,i8,i9,i10,k, n[1+2], n0[1+2];
    double x[1+2],y[1+2],ss[1+2],s,s1,s2,s3,s4,s5,min,x1,x2,y1,y2,xx,yy;
xx=5.0;x[1]=0.0; x[2]=2.0;x[3]=0.0;x[4]=2.0;x[5]=1.0;x[6]=0.5;x[7]=2.5;x[8]=0.5;x[9]=2.5;
//x[10]=1.5;
yy=5.0;y[1]=0.0;y[2]=0.0;y[3]=2.0;y[4]=2.0;y[5]=4.0; y[6]=-0.5; y[7]=0.0; y[8]=1.5; y[9]=2.0;
//y[10]=3.5;
    min=10000.0;
    FILE*file;

    file=fopen("1.txt","w");
    for(i1=1;i1<=l;i1++)
    for(i2=1;i2<=l;i2++)
    for(i3=1;i3<=l;i3++)
    for(i4=1;i4<=l;i4++)
    for(i5=1;i5<=l;i5++)
    for(i6=1;i6<=l;i6++)
    for(i7=1;i7<=l;i7++)
    for(i8=1;i8<=l;i8++)
    for(i9=1;i9<=l;i9++)
    {
```

```

        s=0.0;
x1=x[i1];y1=y[i1];
ss[1]=r(x1,y1,xx,yy);n[1]=i1;
s=s+ss[1];
if( !(i2==i1))
    {
        x1=x[i1];y1=y[i1]; x2=x[i2];y2=y[i2];
        ss[2]=r(x1,y1,x2,y2);n[2]=i2;
        s=s+ss[2];
        if( !(i3==i1) && !(i3==i2))
            {
x1=x[i2];y1=y[i2]; x2=x[i3];y2=y[i3];
        ss[3]=r(x1,y1,x2,y2);n[3]=i3;
        s=s+ss[3];
        if( !(i4==i1) && !(i4==i2) && !(i4==i3))
            {
x1=x[i3];y1=y[i3]; x2=x[i4];y2=y[i4];
        ss[4]=r(x1,y1,x2,y2);n[4]=i4;
        s=s+ss[4];
        if( !(i5==i1) && !(i5==i2) && !(i5==i3) && !(i5==i4))
            {
x1=x[i4];y1=y[i4]; x2=x[i5];y2=y[i5];
        ss[5]=r(x1,y1,x2,y2);n[5]=i5;
        s=s+ss[5];
        if( !(i6==i1) && !(i6==i2) && !(i6==i3) && !(i6==i4) && !(i6==i5))
            {
x1=x[i5];y1=y[i5]; x2=x[i6];y2=y[i6];
        ss[6]=r(x1,y1,x2,y2);n[6]=i6;
        s=s+ss[6];
        if( !(i7==i1) && !(i7==i2) && !(i7==i3) && !(i7==i4) && !(i7==i5) && !(i7==i6) )
            {
x1=x[i6];y1=y[i6]; x2=x[i7];y2=y[i7];
        ss[7]=r(x1,y1,x2,y2);n[7]=i7;
        s=s+ss[7];
        if( !(i8==i1) && !(i8==i2) && !(i8==i3) && !(i8==i4) && !(i8==i5) && !(i8==i6)
&& !(i8==i7) )
            {
x1=x[i7];y1=y[i7]; x2=x[i8];y2=y[i8];
        ss[8]=r(x1,y1,x2,y2);n[8]=i8;
        s=s+ss[8];
        if( !(i9==i1) && !(i9==i2) && !(i9==i3) && !(i9==i4) && !(i9==i5) && !(i9==i6)
&& !(i9==i7) && !(i9==i8) )
            {
x1=x[i8];y1=y[i8]; x2=x[i9];y2=y[i9];
        ss[9]=r(x1,y1,x2,y2);n[9]=i9;
        s=s+ss[9];
        if(s<=min)
            {
min=s;
n0[1]=i1;n0[2]=i2; n0[3]=i3; n0[4]=i4;n0[5]=i5;

```

```

n0[6]=i6; n0[7]=i7; n0[8]=i8; n0[9]=i9;
printf("%d %d %d %d %d %d %d %d %d %d s min=%.16lf\n", i1, i2, i3, i4, i5, i6, i7, i8, i9, min);
fprintf(file, "%d %d %d %d %d %d %d %d %d %d \n", i1, i2, i3, i4, i5, i6, i7, i8, i9);
    }
}}}}}}}} }
    for(k=1; k<=l; k++)
    {
printf("%d %.16lf %.16lf \n", k, x[n0[k]], y[n0[k]]);
fprintf(file, "%.16lf %.16lf \n", x[n0[k]], y[n0[k]]);
    }
}}

```

Последовательность записи программой оптимального (минимального по длине графа) на данный момент времени, последовательность прохождения вершин и суммарная длина ребер графа указана в каждой строке ниже.

```

1 2 3 4 5 6 7 8 9 s min=27.2863611087978240
1 2 3 4 5 6 7 9 8 s min=26.7863611087978240
1 2 3 4 5 6 8 9 7 s min=26.7248082959889930
1 2 3 4 5 7 6 8 9 s min=26.5306704123878810
1 2 3 4 5 7 9 8 6 s min=26.4691175995790500
1 2 3 4 5 8 6 7 9 s min=24.7466254837166770
1 2 3 4 5 9 7 6 8 s min=24.6971157269202840
1 2 3 4 7 6 8 9 5 s min=24.5841533750381560
1 2 3 4 8 6 7 9 5 s min=24.0421865795046830
1 2 3 4 9 5 8 6 7 s min=23.5105575062168910
1 2 3 4 9 7 6 8 5 s min=23.0105575062168910
1 2 3 5 4 9 7 6 8 s min=22.9331837044200770
1 2 3 5 8 4 9 7 6 s min=22.8277643138008660
1 2 3 5 9 4 8 6 7 s min=22.7782545570044730
1 2 3 8 5 4 9 7 6 s min=21.9537322649032270
1 2 3 8 6 7 4 9 5 s min=21.7297073434158750
1 2 3 8 6 7 9 4 5 s min=21.4042225081068340
1 2 4 9 5 3 8 6 7 s min=21.0757953833606420
1 2 4 9 7 6 8 3 5 s min=20.5757953833606420
1 2 6 7 4 9 8 3 5 s min=20.2800398390624930
1 2 6 7 9 4 8 3 5 s min=19.7380730435290200
1 2 7 4 9 5 3 8 6 s min=19.5757953833606420
1 2 7 6 3 8 4 9 5 s min=19.4703759927414380
1 2 7 6 8 3 4 9 5 s min=19.3397274058608560
1 2 7 6 8 3 5 4 9 s min=19.3118633608604320
1 2 7 9 4 5 3 8 6 s min=19.2503105480516010
1 3 8 5 4 9 7 2 6 s min=19.1448911574323940
1 3 8 6 2 7 4 9 5 s min=18.9208662359450410
1 3 8 6 2 7 9 4 5 s min=18.5953814006360040
1 6 2 7 3 8 4 9 5 s min=18.3491211531233740
1 6 2 7 4 9 3 8 5 s min=18.1774827739279840
1 6 2 7 4 9 5 3 8 s min=17.8640409946313810
1 6 2 7 4 9 8 3 5 s min=17.4255938074402120
1 6 2 7 9 4 8 3 5 s min=16.8836270119067390
2 7 4 9 5 3 8 1 6 s min=16.6239250776112040
2 7 6 1 3 8 4 9 5 s min=16.3878571001114180
2 7 6 1 8 3 4 9 5 s min=16.3878571001114180
2 7 6 1 8 3 5 4 9 s min=16.3599930551109980
2 7 9 4 5 3 8 1 6 s min=16.2984402423021630

```

```

3 8 1 6 2 7 4 9 5 s min=15.9689959301956050
3 8 1 6 2 7 9 4 5 s min=15.6435110948865650
4 9 2 7 1 6 8 3 5 s min=15.4544750398010000
4 9 2 7 6 1 3 8 5 s min=15.3294696319064340
4 9 2 7 6 1 8 3 5 s min=14.5971666826940200
4 9 5 3 8 1 6 2 7 s min=14.5551998871605480
4 9 7 2 6 1 8 3 5 s min=14.0551998871605490
5 3 8 1 6 2 7 4 9 s min=13.9972176384677560
5 3 8 1 6 2 7 9 4 s min=13.9356648256589250
5 3 8 4 9 7 2 6 1 s min=13.9356648256589250
5 4 9 7 2 6 1 8 3 s min=13.9356648256589250
9 4 7 2 6 1 8 3 5 s min=13.7792368508034220
1 2.5000000000000000 2.0000000000000000
2 2.0000000000000000 2.0000000000000000
3 2.5000000000000000 0.0000000000000000
4 2.0000000000000000 0.0000000000000000
5 0.5000000000000000 -0.5000000000000000
6 0.0000000000000000 0.0000000000000000
7 0.5000000000000000 1.5000000000000000
8 0.0000000000000000 2.0000000000000000
9 1.0000000000000000 4.0000000000000000
Press any key to continue

```

В последних строках указана последовательность вершин графа и координаты каждой вершины в оптимальной последовательности. Программа генерирует txt файл, из которого можно скопировать координаты вершин конечной оптимальной траектории для построения графика. Из таблицы видно, что исходный путь почтальона можно сократить в 2 раза.

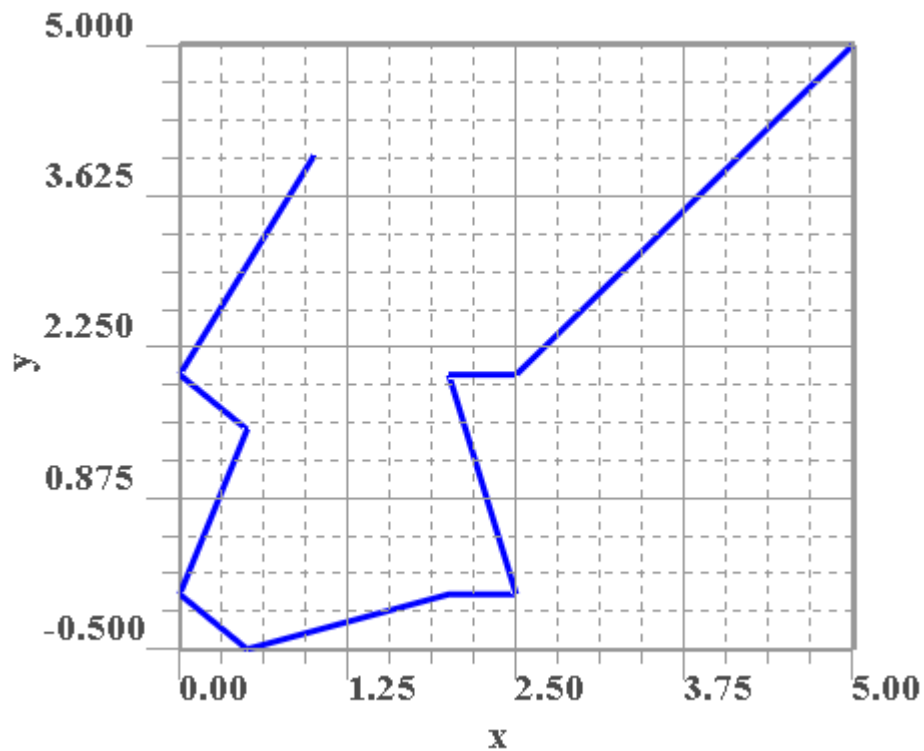


Рис.1. Оптимальная незамкнутая траектория почтальона (терапевта) для связи всех адресатов (пациентов). Исходное положение почтальона (терапевта) с координатами (5.0;5.0)

Глава 2. Математические преобразования над точечными множествами

Практическая работа 4. Слабый и сильный критерии отделимости двух точечных множеств на плоскости

Рассмотрим два точечных множества на плоскости, заданных своими координатами

$$(x_i^1, y_i^1), i = \overline{1, n_1}, (x_i^2, y_i^2), i = \overline{1, n_2} \quad (1)$$

Найдем координаты центров тяжести первого и второго множеств

$$x_{01} = \frac{1}{n_1} \sum_{i=1}^{n_1} x_i^1, y_{01} = \frac{1}{n_1} \sum_{i=1}^{n_1} y_i^1, x_{02} = \frac{1}{n_2} \sum_{i=1}^{n_2} x_i^2, y_{02} = \frac{1}{n_2} \sum_{i=1}^{n_2} y_i^2 \quad (2)$$

Перенесем локальные системы в центр тяжести каждого множества, то есть, центрируем точки каждого множества по формулам

$$x_{1i} = x_i^1 - x_{01}, y_{1i} = y_i^1 - y_{01} (i = \overline{1, n_1}), x_{2i} = x_i^2 - x_{02}, y_{2i} = y_i^2 - y_{02} (i = \overline{1, n_2}) \quad (3)$$

Далее найдем радиус каждого точечного множества, другими словами, найдем максимально удаленные точки от центра тяжести каждого множества. Радиусы множеств конечны, если каждые две точки множества находятся на конечном расстоянии друг от друга (мы рассматриваем конечные точечные множества). Отметим, что все точки множества находятся внутри или на границе окружности данного множества.

$$R_1 = \max_{i=1, n_1} \rho_{1i} = \max_{i=1, n_1} \sqrt{x_{1i}^2 + y_{1i}^2}, R_2 = \max_{i=1, n_2} \rho_{2i} = \max_{i=1, n_2} \sqrt{x_{2i}^2 + y_{2i}^2} \quad (4)$$

Сильный критерий отделимости двух множеств описывается правилом и неравенством: расстояние между центрами множеств не менее суммы радиусов двух множеств. Допускается касание окружностей двух множеств, смотрим рисунок 1.

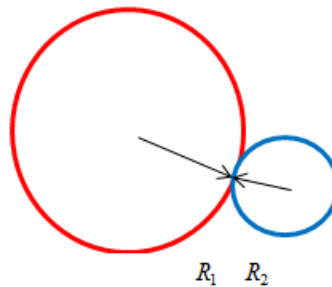


Рис.1. Взаимное расположение окружностей для сильного критерия отделимости множеств

$$R \geq R_1 + R_2, R = \sqrt{(x_{02} - x_{01})^2 + (y_{02} - y_{01})^2} \quad (5)$$

Слабый критерий отделимости двух множеств описывается правилом и неравенством: расстояние между центрами множеств не менее максимума из радиусов двух множеств. Допускается расположение центра окружности первой окружности минимального радиуса на окружности второй окружности максимального радиуса. Действительно, если более половины точек первого множества скрывает область круга второго множества, то первое множество перестает выделяться на фоне большего множества, смотрим рисунок 2.

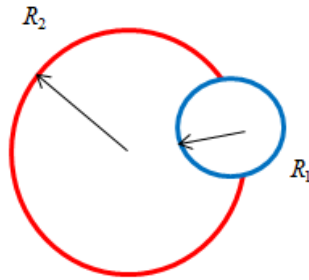


Рис.2. Взаимное расположение окружностей для слабого критерия отделимости множеств

$$R \geq \max \{R_1, R_2\} \Leftrightarrow R \geq \frac{|R_2 - R_1| + R_1 + R_2}{2} \quad (6)$$

В программе на С++(популярный язык программирования в настоящее время) используется именно неравенство (6) для вычисления максимального радиуса окружностей.

```
#include<stdio.h>
#include<math.h>
int const n1=5,n2=10;
int main()
{int i,j,k;
double x1[n1+1],x2[n2+1],y1[n1+1],y2[n2+1],r1,r2,r,rr,x01,x02,y01,y02;
double s1,s2,s3,s4,s5,s6,s7,s8;
for(i=0;i<=n1;i++)
{
x1[i]=0.0;y1[i]=2.0*double(i);
}
for(i=0;i<=n2;i++)
{
x2[i]=double(i);y2[i]=0.0;
}
s1=0.0;s2=0.0;s3=0.0;s4=0.0;s5=0.0;s6=0.0;
for(i=0;i<=n1;i++)
{
s1=s1+x1[i];
s2=s2+y1[i];
}
x01=s1/double(n1+1);y01=s2/double(n1+1);
for(i=0;i<=n2;i++)
{
s3=s3+x2[i];
s4=s4+y2[i];
}
x02=s3/double(n2+1);y02=s4/double(n2+1);

r1=-1.0;
for(i=0;i<=n1;i++)
{ s7=sqrt((x1[i]-x01)*(x1[i]-x01)+(y1[i]-y01)*(y1[i]-y01));
if(s7>=r1)
{
r1=s7;
}
}
}
```

```

r2=-1.0;
for(i=0;i<=n2;i++)
{ s8=sqrt((x2[i]-x02)*(x2[i]-x02)+(y2[i]-y02)*(y2[i]-y02));
if(s8>=r2)
{
r2=s8;
}
}
r=sqrt((x02-x01)*(x02-x01)+(y02-y01)*(y02-y01));
rr=(r1+r2+sqrt((r2-r1)*(r2-r1)))/2.0;
//////////критерий
if(r<r1+r2)
{
printf(" strong kriterij-not\n");
}
if(r>=r1+r2)
{printf("strong kriterii-yes\n");
}
if(r>=rr)
{
printf("weak kriterii-yes\n");
}
if(r<rr)
{
printf("weak kriterii-not\n");
}
}

```

Применим три различных ориентации точечных множеств для тестирования в программе

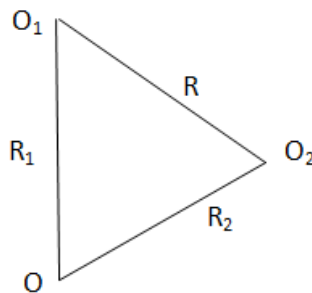


Рис.3. Предельный вариант выполнения слабого критерия отделимости двух точечных множеств. O -точка пересечения двух окружностей одинакового радиуса равному расстоянию между центрами окружностей O_1, O_2 $R_1=R_2=R$. В предельном случае угол между радиусами окружностей $OO_1=R_1$ $OO_2=R_2$ составляет 60° . Если этот угол меньше 60° , то не выполнен ни сильный ни слабый критерии, если угол превышает 60° , то выполнен слабый критерий, а сильный нет. Наконец, если угол равен 180° , то выполнен сильный критерий тоже и окружности равного радиуса касаются друг друга внешним образом.

Пример 1. 6 точек первого множества расположены на оси Oy с интервалом 2 вверх по оси, первая точка совпадает с началом координат, точки расположены на отрезке длиной 10. Второе множество из 11 точек расположены горизонтально вдоль оси Ox , первая точка совпадает с началом координат, остальные с интервалом 1 лежат на отрезке длиной 10. Угол между отрезками $OO_1=R_1$ $OO_2=R_2$ составляет 90° , то есть выполняется слабый критерий, но не выполнен сильный критерий. Программа дает результат

strong kriterij-not

weak kriterii-yes

Press any key to continue

Пример 2. 6 точек первого множества расположены на оси Оу с интервалом 2 вверх по оси, первая точка совпадает с началом координат, точки расположены на отрезке длиной 10. Второе множество из 11 точек расположено вдоль биссектрисы первого координатного угла, первая точка совпадает с началом координат, остальные с интервалом 1 лежат на отрезке длиной 10. Угол между отрезками $OO_1=R_1$ $OO_2=R_2$ составляет 45^0 , то есть не выполняется слабый критерий и не выполнен сильный критерий. Изменена начальная часть программы

```
for(i=0;i<=n1;i++)
```

```
{x1[i]=0.0;y1[i]=2.0*double(i);}
```

```
for(i=0;i<=n2;i++)
```

```
{x2[i]=double(i)/sqrt(2.0);y2[i]=double(i)/sqrt(2.0);}
```

Программа дает результат

strong kriterij-not

weak kriterii-not

Press any key to continue

Пример 3. 6 точек первого множества расположены на оси Оу с интервалом 2 вверх по оси, первая точка совпадает с началом координат, точки расположены на отрезке длиной 10. Второе множество из 11 точек расположены противоположно оси Оу, первая точка совпадает с началом координат, остальные с интервалом 1 лежат на отрезке длиной 10. Угол между отрезками $OO_1=R_1$ $OO_2=R_2$ составляет 180^0 , то есть выполняется слабый критерий и выполнен сильный критерий. Изменена начальная часть программы

```
for(i=0;i<=n1;i++)
```

```
{x1[i]=0.0;y1[i]=2.0*double(i);}
```

```
for(i=0;i<=n2;i++)
```

```
{x2[i]=0.0;y2[i]=-double(i);}
```

Программа дает результат

strong kriterii-yes

weak kriterii-yes

Press any key to continue

Практическая работа 5. Слабый и сильный критерии отделимости двух точечных множеств в пространстве

Рассмотрим два точечных множества на плоскости, заданных своими координатами

$$(x_i^1, y_i^1, z_i^1), i = \overline{1, n_1}, (x_i^2, y_i^2, z_i^2), i = \overline{1, n_2} \quad (1)$$

Найдем координаты центров тяжести первого и второго множеств

$$x_{01} = \frac{1}{n_1} \sum_{i=1}^{n_1} x_i^1, y_{01} = \frac{1}{n_1} \sum_{i=1}^{n_1} y_i^1, z_{01} = \frac{1}{n_1} \sum_{i=1}^{n_1} z_i^1, x_{02} = \frac{1}{n_2} \sum_{i=1}^{n_2} x_i^2, y_{02} = \frac{1}{n_2} \sum_{i=1}^{n_2} y_i^2, z_{02} = \frac{1}{n_2} \sum_{i=1}^{n_2} z_i^2 \quad (2)$$

Перенесем локальные системы в центр тяжести каждого множества, то есть, центрируем точки каждого множества по формулам

$$x_{1i} = x_i^1 - x_{01}, y_{1i} = y_i^1 - y_{01}, z_{1i} = z_i^1 - z_{01} \quad (i = \overline{1, n_1}), x_{2i} = x_i^2 - x_{02}, y_{2i} = y_i^2 - y_{02}, z_{2i} = z_i^2 - z_{02} \quad (i = \overline{1, n_2}) \quad (3)$$

Далее найдем радиус каждого точечного множества, другими словами, найдем максимально удаленные точки от центра тяжести каждого множества. Радиусы множеств конечны, если каждые две точки множества находятся на конечном расстоянии друг от друга (мы рассматриваем конечные точечные множества). Отметим, что все точки множества находятся внутри или на поверхности шара данного множества.

$$R_1 = \max_{i=1,n_1} \rho_{1i} = \max_{i=1,n_1} \sqrt{x_{1i}^2 + y_{1i}^2 + z_{1i}^2}, R_2 = \max_{i=1,n_2} \rho_{2i} = \max_{i=1,n_2} \sqrt{x_{2i}^2 + y_{2i}^2 + z_{2i}^2} \quad (4)$$

Сильный критерий *отделимости* двух множеств описывается правилом и неравенством: расстояние между центрами множеств не менее суммы радиусов двух множеств. Допускается касание шаров двух множеств, смотрим рисунок 1.

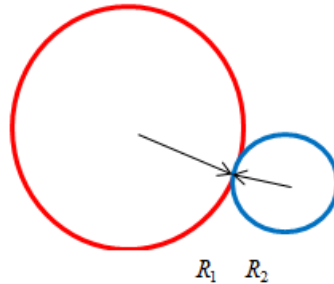


Рис.1. Взаимное расположение шаров для сильного критерия отделимости множеств

$$R \geq R_1 + R_2, R = \sqrt{(x_{02} - x_{01})^2 + (y_{02} - y_{01})^2} \quad (5)$$

Слабый критерий отделимости двух множеств описывается правилом и неравенством: расстояние между центрами множеств не менее максимума из радиусов двух множеств. Допускается расположение центра первого шара минимального радиуса на сфере второго шара максимального радиуса. Действительно, если более половины точек первого множества поглощает шар второго множества, то первое множество перестает выделяться на фоне большего множества, смотрим рисунок 2.

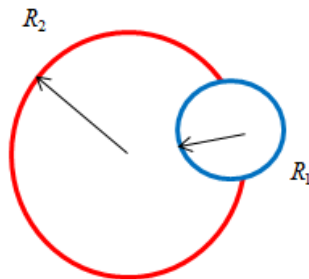


Рис.2. Взаимное расположение шаров для слабого критерия отделимости множеств

$$R \geq \max \{R_1, R_2\} \Leftrightarrow R \geq \frac{|R_2 - R_1| + R_1 + R_2}{2} \quad (6)$$

В программе на С++ используется неравенство (6) для вычисления максимального радиуса окружностей.

```
#include "stdafx.h"
#include<stdio.h>
#include<math.h>
int const n1=5,n2=10;
int main()
{ int i,j,k;
double x1[n1+1],x2[n2+1],y1[n1+1],y2[n2+1],r1,r2,r,rr,x01,x02,y01,y02;
double s1,s2,s3,s4,s5,s6,s7,s8,s9,s10,s11,s12,z02,z01,z1[n1+1],z2[n2+1];
for(i=0;i<=n1;i++)
{
x1[i]=0.0;y1[i]=2.0*double(i);z1[i]=0.0;
}
```

```

for(i=0;i<=n2;i++)
{
x2[i]=double(i);y2[i]=0.0;z2[i]=0.0;
}
s1=0.0;s2=0.0;s3=0.0;s4=0.0;s5=0.0;s6=0.0;
for(i=0;i<=n1;i++)
{
s1=s1+x1[i];s2=s2+y1[i];s3=s3+z1[i];
}
x01=s1/double(n1+1);y01=s2/double(n1+1);z01=s3/double(n1+1);
for(i=0;i<=n2;i++)
{
s4=s4+x2[i];s5=s5+y2[i];s6=s6+z2[i];
}
x02=s4/double(n2+1);y02=s5/double(n2+1);z02=s6/double(n2+1);

r1=-1.0;
for(i=0;i<=n1;i++)
{ s7=sqrt((x1[i]-x01)*(x1[i]-x01)+(y1[i]-y01)*(y1[i]-y01)+(z1[i]-z01)*(z1[i]-z01));
if(s7>=r1)
{
r1=s7;
}
}
r2=-1.0;
for(i=0;i<=n2;i++)
{ s8=sqrt((x2[i]-x02)*(x2[i]-x02)+(y2[i]-y02)*(y2[i]-y02)+(z2[i]-z02)*(z2[i]-z02));
if(s8>=r2)
{
r2=s8;
}
}
r=sqrt((x02-x01)*(x02-x01)+(y02-y01)*(y02-y01)+(z02-z01)*(z02-z01));
rr=(r1+r2+sqrt((r2-r1)*(r2-r1)))/2.0;
//////////критерий
if(r<r1+r2)
{
printf(" strong kriterij-not\n");
}
if(r>=r1+r2)
{printf("strong kriterii-yes\n");
}
if(r>=rr)
{
printf("weak kriterii-yes\n");
}
if(r<rr)
{printf("weak kriterii-not\n");}
}

```

Применим три различных ориентации точечных множеств для тестирования программы

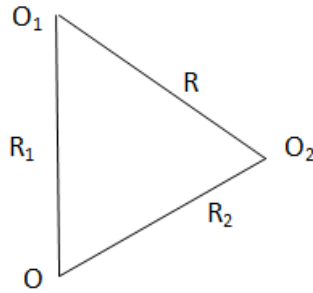


Рис.3. Предельный вариант выполнения слабого критерия делимости двух точечных множеств. \$O\$-точка пересечения двух сфер одинакового радиуса равному расстоянию между центрами сфер \$O_1, O_2\$ \$R_1=R_2=R\$. В предельном случае угол между радиусами сфер \$OO_1=R_1\$ \$OO_2=R_2\$ составляет \$60^\circ\$. Если этот угол меньше \$60^\circ\$, то не выполнен ни сильный, ни слабый критерии, если угол превышает \$60^\circ\$, то выполнен слабый критерий, а сильный нет. Наконец, если угол равен \$180^\circ\$, то выполнен сильный критерий тоже, а сферы равного радиуса касаются друг друга внешним образом.

Пример 1. 6 точек первого множества расположены на оси \$Oy\$ с интервалом 2 вверх по оси, первая точка совпадает с началом координат, точки расположены на отрезке длиной 10. Второе множество из 11 точек расположены горизонтально вдоль оси \$Ox\$, первая точка совпадает с началом координат, остальные с интервалом 1 лежат на отрезке длиной 10. Угол между отрезками \$OO_1=R_1\$ \$OO_2=R_2\$ составляет \$90^\circ\$, то есть выполняется слабый критерий, но не выполнен сильный критерий. Программа дает результат

```
strong kriterij-not
weak kriterii-yes
Press any key to continue
```

Пример 2. 6 точек первого множества расположены на оси \$Oy\$ с интервалом 2 вверх по оси, первая точка совпадает с началом координат, точки расположены на отрезке длиной 10. Второе множество из 11 точек расположены вдоль высоты правильной равносторонней пирамиды, первая точка совпадает с началом координат, остальные с интервалом 1 лежат на отрезке длиной 10. Угол между отрезками \$OO_1=R_1\$ \$OO_2=R_2\$ равен

$$\gamma = \arccos \left(\sqrt{1 - (2/3)^2 \left(\sqrt{1 - (1/2)^2} \right)^2} / 1 \right) = \arccos \sqrt{1 - \frac{4}{9} \cdot \frac{3}{4}} = \arccos \sqrt{\frac{2}{3}} = 0.615480 \dots$$

$$\pi/3 = 1.04719755 \dots$$

то есть, меньше \$60^\circ\$, поэтому не выполняется слабый критерий и не выполнен сильный критерий. Изменена начальная часть программы

```
for(i=0;i<=n1;i++)
{x1[i]=0.0;y1[i]=double(i);z1[i]=0.0;}
for(i=0;i<=n2;i++)
{x2[i]=double(i)*sqrt(2.0/3.0);y2[i]=x2[i];z2[i]=x2[i];}
```

Программа дает результат

```
strong kriterij-not
weak kriterii-not
Press any key to continue
```

Пример 3. 6 точек первого множества расположены на оси \$Oy\$ с интервалом 2 вверх по оси, первая точка совпадает с началом координат, точки расположены на отрезке длиной 10. Второе множество из 11 точек расположены противоположно оси \$Oy\$, первая точка совпадает с началом координат, остальные с интервалом 1 лежат на отрезке длиной 10. Угол между отрезками \$OO_1=R_1\$ \$OO_2=R_2\$ составляет \$180^\circ\$, то есть выполняется слабый критерий и выполнен сильный критерий. Изменена начальная часть программы

```
for(i=0;i<=n1;i++)
```

```

{x1[i]=0.0;y1[i]=double(i);z1[i]=0.0;}
for(i=0;i<=n2;i++)
{x2[i]=0.0;y2[i]=-double(i);z2[i]=0.0;}
Программа дает результат
strong kriterii-yes
weak kriterii-yes
Press any key to continue

```

Практическая работа 6. Задача поиска главных направлений точечного множества на плоскости

Пусть заданы координаты n точек на плоскости $(x_i, y_i), i = \overline{1, n}$ (1)

Найдем координаты центра тяжести множества и координаты точек относительно центра тяжести $\bar{x}_i, \bar{y}_i, i = \overline{1, n}$

$$x_0 = \frac{1}{n} \sum_{i=1}^n x_i, y_0 = \frac{1}{n} \sum_{i=1}^n y_i, \bar{x}_i = x_i - x_0, \bar{y}_i = y_i - y_0, i = \overline{1, n} \quad (2)$$

В работе [245] показано, что главные направления прямой, проходящей через центр тяжести точечного множества на плоскости под углом φ к оси абсцисс определяются формулами

$$\tan(\varphi) = \frac{A - C \pm \sqrt{(A - C)^2 + 4B^2}}{2B}, \varphi_{1,2} = \arctan\left(\frac{A - C \pm \sqrt{(A - C)^2 + 4B^2}}{2B}\right) \quad (3)$$

$$\text{Где: } A = \sum_{i=1}^n \bar{x}_i^2, B = \sum_{i=1}^n \bar{x}_i \bar{y}_i, C = \sum_{i=1}^n \bar{y}_i^2$$

Первое направление (угол φ_1) определяет минимум суммы квадратов расстояний всех точек до выделенной прямой, а второй угол φ_2 дает максимум суммы квадратов расстояний всех точек до прямой.

Используя формулы (1), (2), (3), запишем код программы на языке C++ и приведем пример, тестирующий данную программу для самого неблагоприятного случая.

```

#include "stdafx.h"
#include <stdio.h>
#include <math.h>
const int n = 8;
int main() {
    double A,B,C,X[n + 1], Y[n + 1], pi,x0, y0;
    double x1[n + 1], y1[n + 1], alfa1, alfa2, S1, S2, S3, S4, S5, S6, S7;
    int i;
    X[1] = -2.0; Y[1] = 1.0+0.0000000000000001;
    X[2] = 0.0; Y[2] = 1.0;
    X[3] = 2.0; Y[3] = 1.0;
    X[4] = 2.0; Y[4] = 0.0;
    X[5] = 2.0; Y[5] = -1.0;
    X[6] = 0.0; Y[6] = -1.0;
    X[7] = -2.0; Y[7] = -1.0;
    X[8] = -2.0; Y[8] = 0.0;
    S1 = 0.0; S2 = 0.0;pi=2.0*asin(1.0);
    for (i = 1; i <= n; i++) {
        S1 = S1+X[i];

```

```

    S2 = S2+Y[i];
}
x0 = S1 / double(n);
y0 = S2 / double(n);
for (i = 1; i <= n; i++) {
    x1[i] = X[i] - x0;
    y1[i] = Y[i] - y0;
}
S3 = 0.0; S4 = 0.0; S5=0.0;
for (i = 1; i <= n; i++) {
    S3 = S3+x1[i] * x1[i];
    S4 = S4+x1[i] * y1[i];
    S5 = S5+y1[i] * y1[i];
}
A = S3;
B = S4;
C=S5;
S5 = sqrt((A-C) * (A-C) + 4.0 * B * B);
S6 = ((A-C)+S5) / (2.0 * B);
S7 = ((A-C)-S5) / (2.0 * B);
alfa1 = atan(S6);
alfa2 = atan(S7);
printf("alfa1 = %.16lf\n", alfa1);
printf("alfa2 = %.16lf\n", alfa2);
printf("pi/2 = %.16lf\n", pi/2.0);
return 0;
}

```

Замечание 1. Для симметричного расположения точек, например, для тестирующего примера 8 точек расположены центрально симметрично относительно центра масс множества (начала системы координат) коэффициент В в формуле (3) будет равен нулю. Так как точки расположены в вершинах прямоугольного прямоугольника со сторонами 4×2 (центр симметрии расположен в начале координат) и на пересечении сторон с осями координат. Чтобы исключить случай равенства нулю коэффициента В, к одной из координат 8 точек добавлено малое положительное число 0.000000000000001, то есть симметричное множество точек сделаем чуть ассиметричным. Этого достаточно, чтобы вычислить любой угол по формуле (3) с четырнадцатью верными знаками. Для координат 8 точек приведенных в тексте программа дает результат

```

alfa1 = -1.5707963267948954
alfa2 = 0.0000000000000000
pi/2 = 1.5707963267948966
Press any key to continue

```

Замечание 2. Вычислим явно сумму квадратов расстояний 8 указанных в программе точек до экстремальных прямых. Если прямая расположена горизонтально и проходит через начало координат, получим $E_1=6 \times 1^2=6$. Если рассмотреть вертикальную прямую, имеем $E_2=6 \times 2^2=24$. Поэтому в формуле (3) знак плюс перед радикалом соответствует максимальному значению функции ошибок E_2 , а знак минус в формуле (3) перед радикалом соответствует минимальному значению функции ошибок, то есть, E_1 .

Глава 3. Несколько вариантов математического описания однослойной нейронной сети

Лекция 2. Задача первая. Прямоугольная матрица плюс вектор весовых коэффициентов на входе нейронного монослоя. Практическая работа 7.

Рассмотрим следующую математическую модель. Вектор входного сигнала представлен в двоичном коде и принимает всего два значения $\{1, -1\}$ и имеет размерность n . Данное множество из двух чисел является аналогом множества остатков целых чисел по модулю два $\{1, 0\}$. Вектор сигнала, принимаемый нейронами, имеет другую размерность m и равен

$$y = Cx + b, \quad y_i = \sum_{j=1}^n C_{i,j} x_j + b_i, \quad i = \overline{1, m}, \quad x_j = \{-1, 1\}, \quad j = \overline{1, n} \quad (2.1)$$

В формуле (2.1) $C_{i,j}, i = \overline{1, m}, j = \overline{1, n}$ – прямоугольная матрица весовых коэффициентов, $b_i, i = \overline{1, m}$ – вектор весовых коэффициентов. В пространстве размерности $m \times (n+1)$ изменением матрицы $C_{i,j}, i = \overline{1, m}, j = \overline{1, n}$ и вектора $b_i, i = \overline{1, m}$ проводится оптимизация задачи, то есть минимизируется функционал ошибки.

На выходе нейронов сигнал отличается от входного сигнала тем, что он обрабатывается функцией активации, которая является математической моделью отклика живого нейрона человеческого мозга на воздействие входного сигнала. Для простоты мы выбрали наиболее распространенную и универсальную, но модифицированную функцию “сигмоиды”:

$$z_i = \overline{\sigma}(y_i), \quad i = \overline{1, m}, \quad \overline{\sigma}(y) = \frac{2}{1 + \exp(-y)} - 1 \in [-1, 1], \quad y \in R \quad (2.2)$$

$$\overline{\sigma}'(y) = \frac{d\overline{\sigma}(y)}{dy} = \frac{2 \exp(-y)}{(1 + \exp(-y))^2} = 2 \frac{\exp(-y)}{(1 + \exp(-y))} \cdot \frac{1}{(1 + \exp(-y))} = 2\sigma(y)(1 - \sigma(y)) \quad (2.3)$$

Где $\sigma(y) = \frac{1}{1 + \exp(-y)} \in [0, 1]$ – классическая функция сигмоиды

В данной простой задаче после цикла оптимизации на выходе каждого нейрона выполняется отображение вектора z_i на множество значений $\{1, -1\}$ по правилу

$$t_i = f(z_i) = \begin{cases} 1, & \text{если } z_i \geq z_{hop} \\ -1, & \text{если } z_i < z_{hop} \end{cases}, \quad i = \overline{1, m} \quad (2.4)$$

В формуле (2.4) z_{hop} – пороговое значение, по умолчанию равно нулю.

В данной и в остальных задачах используется модель обучения нейронной сети “с учителем”. То есть, для всех состояний $N = 2^n$ входного сигнала вектор функции учителя $\overline{z}_i^k, i = \overline{1, m}, k = \overline{0, N-1}$ показывает каким должен быть безошибочным сигнал на выходе нейронной сети.

Таким образом, классическая функция ошибок суммирует квадрат разности функции учителя и выходного сигнала для каждой компоненты выходного вектора и для каждого возможного вектора входного сигнала. В случае если число входных состояний сигнала меньше даже числа переменных весовых коэффициентов в задаче $N = 2^n < m \times (n+1)$, то нужно взять повторяющийся входной сигнал с числом состояний кратному основному $N = p \cdot 2^n, p \in N$. С учетом оптимизации функции ошибок итерационным методом градиентного спуска также кратно числу различных состояний $N = 2^n$ входного сигнала приходится увеличивать число итераций процесса обучения нейронной сети. Запишем функцию ошибок

$$E(C_{i,j}, b_i) = \sum_{k=0}^{N-1} \sum_{i=1}^m \left(z_i^k - \bar{z}_i^{-k} \right)^4 = \sum_{k=0}^{N-1} \sum_{i=1}^m \left(\sigma \left(\sum_{j=1}^n C_{i,j} x_j + b_i \right) - \bar{z}_i^{-k} \right)^4 \quad (2.5)$$

Обратим внимание на то, что функцию принятия решения по формуле (4) мы применяем один раз после того как по циклу численно решена задача поиска минимума функции ошибок (2.5) методом градиентного спуска. Внутри итерационного цикла переменная z_i^k меняется, оставаясь действительным числом, а в конце алгоритма каждое значение $z_i^{N-1}, i = \overline{1, m}$ скачком отображается в число -1 либо в 1. Для того чтобы величина градиента была достаточно большой и уменьшить число итераций в функции ошибок суммируются не квадраты невязок задачи, а четвертые степени невязок.

Найдем компоненты градиента функции ошибок

$$\begin{aligned} \frac{\partial E(C_{i,j}, b_i)}{\partial C_{i,j}} &= \sum_{k=0}^{N-1} 4 \left(\sigma \left(\sum_{j=1}^n C_{i,j} x_j + b_i \right) - \bar{z}_i^{-k} \right)^3 2\sigma(y_i)(1-\sigma(y_i))x_j = \\ &= 8 \sum_{k=0}^{N-1} \left(\sigma \left(\sum_{j=1}^n C_{i,j} x_j + b_i \right) - \bar{z}_i^{-k} \right)^3 \sigma(y_i)(1-\sigma(y_i))x_j, y_i = \sum_{j=1}^n C_{i,j} x_j + b_i, i = \overline{1, m}, j = \overline{1, n} \quad (2.6) \end{aligned}$$

Замечание 1. Именно благодаря тому, что в формуле (2.6)

$$x_j \in \{-1, 1\}, \sigma(y_i) > 0, 1 - \sigma(y_i) > 0 \quad \frac{\partial E(C_{i,j}, b_i)}{\partial C_{i,j}} = 0 \Leftrightarrow \sigma \left(\sum_{j=1}^n C_{i,j} x_j + b_i \right) - \bar{z}_i^{-k} = 0, \text{ что равносильно}$$

тому, что во всех случаях выходной сигнал совпадает с функцией учителя. Иначе, если $x_j = 0$, мы получим равенство нулю компоненты градиента даже в случае, когда выходной сигнал не совпадает со значением функции учителя в точке $x_j = 0$. То есть, имеем "фиктивные", не настоящие точки локального минимума функции ошибок (2.5).

Сравнение формул (2.5) и (2.6) показывает, что в формуле (2.6) нет суммирования по индексам i и частично по индексу j (в конце (2.6)), так как данные индексы содержит коэффициент $C_{i,j}$. Аналогично получим производные

$$\frac{\partial E(C_{i,j}, b_i)}{\partial b_i} = \sum_{k=0}^{N-1} 4 \left(\sigma \left(\sum_{j=1}^n C_{i,j} x_j + b_i \right) - \bar{z}_i^{-k} \right)^3 2\sigma(y_i)(1-\sigma(y_i)), i = \overline{1, m} \quad (2.7)$$

В итоге получим формулы для алгоритма градиентного спуска

$$\begin{cases} C_{i,j}^{s+1} = C_{i,j}^s - h \frac{\partial E(C_{i,j}^s, b_i^s)}{\partial C_{i,j}}, s = 1, 2, \dots, N_1, N_1 = p2^n, j = \overline{1, n} \\ b_i^{s+1} = b_i^s - h \frac{\partial E(C_{i,j}^s, b_i^s)}{\partial b_i} \end{cases} \quad (2.8)$$

В формулах (2.8) h – шаг градиентного спуска, а s – номер итерации.

Используем формулы алгоритма (2.6), (2.7), (2.8) для программы на языке Fortran, выбор языка объясняется близким математическим написанием формул, как в тексте работы, так и в коде программы, кроме того, умножение матриц сводится к упрощенной записи $C(:, :) = \text{matmull}(A(:, :), B(:, :))$ и экономии числа вложенных циклов.

```
program net
use dfimsl
use msimsl
```

```

use msflib
integer(8), parameter:: m=4,n=4,n1=15,n2=100
integer(8)::i,j,k,p,xx,yy,x(n),teacher(n,0:n1)
real(8)::cc1(m,n),cc2(m,n),b1(m),b2(m),y(m),x1,g,gg,gg1,porog,x2,s,s1,s2,s3,s4
real(8)::h,e0,mas2(n,0:n1),teacher2(m,0:n1),mas(m,0:n1),z(m),eps,ec(m,n),eb(m)
real(8) status2,xe,ye,m1(0:n1),m2(0:n1),m3(0:n1),res0(m,0:n1)
type(windowconfig) wc
data wc.numxpixels,wc.numypixels,wc.numtextcols, wc.numtextrows, wc.numcolors,
wc.fontSize/6*-1/
g(x1)=1d0/(1d0+dexp(-x1))
gg(x1)=2d0/(1d0+dexp(-x1))-1d0
gg1(x1)=2d0*g(x1)*(1d0-g(x1))
res=setwindowconfig(wc)
res=getwindowconfig(wc)
xe=wc.numxpixels
ye=wc.numypixels
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
print*, " one neuron net"
h1=10d-1;h2=h1;porog=0d0
print*,x1,g(x1),gg(x1),gg1(x1)
do i=1,m
do j=1,n
cc1(i,j)=0d0;cc2(i,j)=0d0;
enddo
enddo
do i=1,m
b1(i)=0d0;b2(i)=0d0;
enddo
do k=0,n1
xx=k;
do i=1,n
yy=mod(xx,2);x(i)=yy
xx=(xx-yy)/2;
teacher(i,k)=2*x(i)-1
mas2(i,k)=dfloat(2*x(i)-1)
enddo
enddo

do k=0,n1
do i=1,m
teacher2(i,k)=mas2(i,n1-int(k/1))
enddo
enddo
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
do p=1,n2

do k=0,n1
y(:)=matmul(cc1(:,p),mas2(:,k))+b1(:)
do i1=1,m
z(i1)=gg(y(i1))
mas(i1,k)=z(i1)
enddo

```

```

enddo

do i=1,m
do j=1,n
s=0d0
s2=0d0
do k=0,n1
x2=mas2(j,k)
x1=mas(i,k)
s=s+2d0*(4d0*gg1(x1)*(x1-teacher2(i,k))**3d0)*x2
s2=s2+2d0*4d0*gg1(x1)*(x1-teacher2(i,k))**3d0
enddo
ec(i,j)=s
enddo
eb(i)=s2
enddo

do i=1,m
b2(i)=b1(i)-h2*eb(i)
enddo
do i=1,m
b1(i)=b2(i)
enddo

do i=1,m
do j=1,n
cc2(i,j)=cc1(i,j)-h1*ec(i,j)
enddo
enddo

do i=1,m
do j=1,n
cc1(i,j)=cc2(i,j)
enddo
enddo

enddo
!!!!!!!!!!!!!!!!!!!!!!proverka
do k=0,n1
y(:)=matmul(cc1(:,:),mas2(:,k))+b1(:)
do i=1,m
s=gg(y(i))
if(s<porog)then
z(i)=-1d0
elseif(s>=porog)then
z(i)=1d0
endif
res0(i,k)=z(i)
enddo
enddo
!!!!!!!!!!!!!!!!!!!!!!funkcia oshibok
eps=0d0;s1=0d0

```

```

do k=0,n1
s=0d0
do i=1,m
s=s+(res0(i,k)-teacher2(i,k))**4d0
s1=s1+s
!print*,res(i,k),teacher2(i,k)
enddo
!print*,"s",s
!pause
if(s<=1d-1)then
eps=eps
elseif(s>1d-1)then
eps=eps+1d0
endif
!pause
enddo
e0=s1
print*,"e0=",e0,"eps=",eps
pause
do i=0,15
m1(i)=dfloat(i)
s1=0d0
do j=1,m
if(teacher2(j,i)<0d0)then
s=0d0
elseif(teacher2(j,i)>0d0)then
s=1d0
endif
s1=s1+s*2d0**dfloat(j-1)
enddo
m2(i)=s1
s3=0d0
do j=1,m
if(res0(j,i)<0d0)then
s4=0d0
elseif(res0(j,i)>0d0)then
s4=1d0
endif
s3=s3+s4*2d0**dfloat(j-1)
enddo
m3(i)=s3
enddo
call axis1(m1)
pause
call axis2(m2)
pause
call axis3(m3)
pause

```

contains

subroutine axis1(m1)

```

type(xycoord) xy
integer(8)::i,j
real(8)::status2,m1(0:n1)
do i=0,n1
status2=setcolor(m1(i)+0_8)
do j=0,10
call moveto(int2(xe/10d0)-10,int2((ye-dfloat(15*i+j))/2),xy)
status2=lineto(1_1*xe/10d0+10_1,(ye-dfloat(15*i+j))/2_2)
enddo
enddo
end subroutine axis1

```

```

subroutine axis2(m2)
type(xycoord) xy
integer(8)::i,j
real(8)::status2,m2(0:n1)
do i=0,n1
status2=setcolor(m2(i)+0_8)
do j=0,10
call moveto(int2((xe)/50)-10,int2((ye-dfloat(15*i+j))/2),xy)
status2=lineto(1_1*(xe)/50+10_1,(ye-dfloat(15*i+j))/2_2)
enddo
enddo
end subroutine axis2

```

```

subroutine axis3(m3)
type(xycoord) xy
integer(8)::i
real(8)::status2,m3(0:n1)
do i=0,n1
status2=setcolor(m3(i)+0_8)
do j=0,10
call moveto(int2((xe)/5)-10,int2((ye-dfloat(15*i+j))/2),xy)
status2=lineto(1_1*(xe)/5+10_1,(ye-dfloat(15*i+j))/2_2)
enddo
enddo
end subroutine axis3
end

```

Рассмотрим задачу преобразования цветности предмета, а также задачу сжатия информации о цветности. В данной задаче и в программе $m=n=4$, число итераций в цикле равно $n_2=100$, номер цвета меняется от 0(черный цвет) до 15(белый цвет). Кроме того, первые цвета с номерами от 0 до 7 являются тусклыми, а от 8 до 15 яркими цветами. Рассмотрим формулу преобразования цветности

$$n_2(i) = N_1 - 1 - \text{mod}(i, n_3) \quad (2.9)$$

- 1) Пример, в центральном столбце исходная последовательность цвета на входе нейронной сети, слева функция учителя с инверсией цвета, справа обученная нейронная сеть

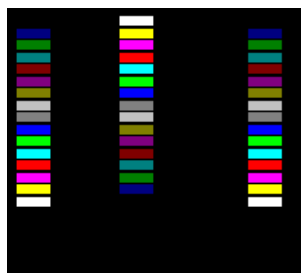


Рис.1. Случай инверсии цветности $n_2(i) = 15 - \text{mod}(i,1) = 15 - i$
 $\epsilon_0 = 0.0000000000000000E+000$ $\text{eps} = 0.0000000000000000E+000$ - программа дает функцию ошибок и число неправильно обученных цветов равных нулю.

2) Пример - задача о сжатии цвета в 2 раза: $n_2(i) = 15 - \text{mod}(i,2)$

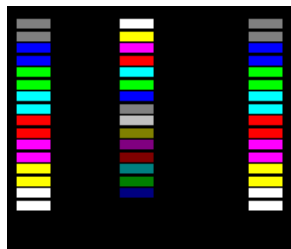


Рис.2. Задача о сжатии цветности в 2 раза. $\epsilon_0 = 0.0000000000000000E+000$ $\text{eps} = 0.0000000000000000E+000$ - программа дает функцию ошибок и число неправильно обученных цветов равных нулю.

3) Пример - задача о сжатии цвета в 3 раза: $n_2(i) = 15 - \text{mod}(i,3)$

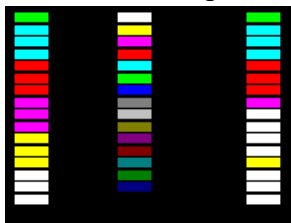


Рис.3. Задача о сжатии цветности в 3 раза. $\epsilon_0 = 224.00000000000000$ $\text{eps} = 4.0000000000000000$ - программа дает функцию ошибок равной 224 и 4 неправильно обученных цвета.

4) Пример - задача о сжатии цвета в 4 раза: $n_2(i) = 15 - \text{mod}(i,4)$

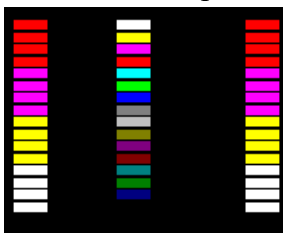


Рис.4. Задача о сжатии цветности в 4 раза. $\epsilon_0 = 0.0000000000000000$ $\text{eps} = 0.0000000000000000$ - программа дает функцию ошибок равной 0 и 0 неправильно обученных цвета.

5) Пример - задача о сжатии цвета в 5 раз: $n_2(i) = 15 - \text{mod}(i,5)$

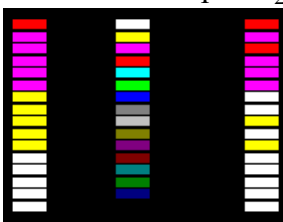


Рис.5. Задача о сжатии цветности в 5 раз. $\epsilon_0 = 256.0000000000000$ $\epsilon_{ps} = 4.000000000000000$ - программа дает функцию ошибок равной 256 и 4 неправильно обученных цвета.

6) Пример - задача о сжатии цвета в 6 раз: $n_2(i) = 15 - \text{mod}(i,6)$

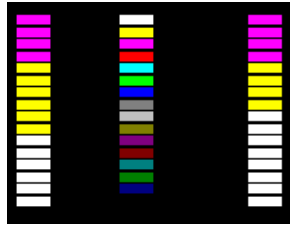


Рис.6. Задача о сжатии цветности в 6 раз. $\epsilon_0 = 128.0000000000000$ $\epsilon_{ps} = 2.000000000000000$ - программа дает функцию ошибок равной 128 и 2 неправильно обученных цвета.

7) Пример - задача о сжатии цвета в 7 раз: $n_2(i) = 15 - \text{mod}(i,7)$

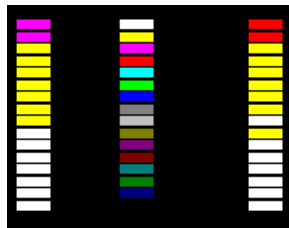


Рис.7. Задача о сжатии цветности в 7 раз. $\epsilon_0 = 256.0000000000000$ $\epsilon_{ps} = 4.000000000000000$ - программа дает функцию ошибок равной 256 и 4 неправильно обученных цвета.

8) Пример - задача о сжатии цвета в 8 раз: $n_2(i) = 15 - \text{mod}(i,8)$

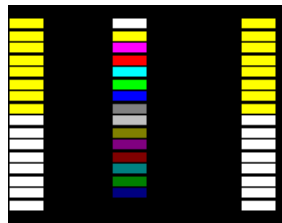


Рис.8. Задача о сжатии цветности в 8 раз. $\epsilon_0 = 0.0000000000000$ $\epsilon_{ps} = 0.000000000000000$ - программа дает функцию ошибок равной 0 и 0 неправильно обученных цвета.

9) Пример - задача о сжатии цвета в 9 раз: $n_2(i) = 15 - \text{mod}(i,9)$

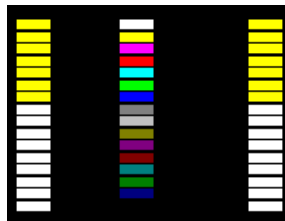


Рис.9. Задача о сжатии цветности в 9 раз. $\epsilon_0 = 0.0000000000000$ $\epsilon_{ps} = 0.000000000000000$ - программа дает функцию ошибок равной 0 и 0 неправильно обученных цвета.

10) Пример - задача о сжатии цвета в 10 раз: $n_2(i) = 15 - \text{mod}(i,10)$

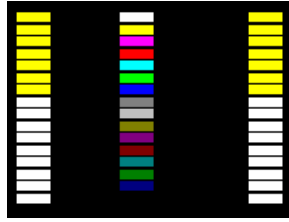


Рис.10. Задача о сжатии цветности в 10 раз. $\epsilon_0 = 0.000000000000$ $\epsilon_{ps} = 0.0000000000000000$ - программа дает функцию ошибок равной 0 и 0 неправильно обученных цвета.

Замечание 2. Анализируя все случаи стопроцентного обучения нейронной сети можно сказать, что они реализуются, когда число состояний входного сигнала (16) делится нацело на модуль (1,2,4,8), а также когда модуль больше 8.

Замечание 3. Из всех случаев обучения нейронной моносети необходимо выбрать худший (75% обучения- 4 неправильных цвета из 16) и взвешенный процент обучения, вычисленный по формуле

$$\bar{E} = \frac{1}{N} \sum_{i=0}^{N-1} n_i p_i = \frac{1}{16} (12 \cdot 1 + 3 \cdot 12/16 + 1 \cdot 14/16) = 0,9063 \approx 90,6\%$$

Замечание 4. В случаях неправильного распознавания сетью цветов, как видно из рисунков 3, 5, 6, 7 цвета на функции учителя и неправильно обученные являются близкими или различаются на 2 позиции, как число пальцев на фотографии президента Нетаньяху (6 вместо 5), которую построила известная программа с нейросетью.

Лекция 3. Задача вторая. Прямоугольная матрица плюс вектор весовых коэффициентов на входе и квадратная матрица на выходе нейронного монослоя. Практическая работа 8.

Рассмотрим следующую математическую модель. Вектор входного сигнала представлен в двоичном коде и принимает всего два значения $\{1, -1\}$ и имеет размерность n . Данное множество из двух чисел является аналогом множества остатков целых чисел по модулю два $\{1, 0\}$. Вектор сигнала, принимаемый нейронами, имеет другую размерность m и равен

$$y = Cx + b, \quad y_s = \sum_{j=1}^n C_{s,j} x_j + b_s, \quad s = \overline{1, m}, \quad x_j = \{-1, 1\}, \quad j = \overline{1, n} \quad (3.1)$$

В формуле (3.1) $C_{s,j}, s = \overline{1, m}, j = \overline{1, n}$ - прямоугольная матрица весовых коэффициентов, $b_s, s = \overline{1, m}$ - вектор весовых коэффициентов. В пространстве размерности $m \times (n+1)$ изменением матрицы $C_{s,j}, s = \overline{1, m}, j = \overline{1, n}$ и вектора $b_s, s = \overline{1, m}$ проводится оптимизация задачи, то есть минимизируется функционал ошибки.

На выходе нейронов сигнал отличается от входного сигнала тем, что он обрабатывается функцией активации, которая является математической моделью отклика живого нейрона человеческого мозга на воздействие входного сигнала. Для простоты мы выбрали наиболее распространенную и универсальную, но модифицированную функцию "сигмоиды":

$$z_s = \bar{\sigma}(y_s), \quad s = \overline{1, m}, \quad \bar{\sigma}(y) = \frac{2}{1 + \exp(-y)} - 1 \in [-1, 1], \quad y \in R \quad (3.2)$$

$$\bar{\sigma}'(y) = \frac{d\bar{\sigma}(y)}{dy} = \frac{2 \exp(-y)}{(1 + \exp(-y))^2} = 2 \frac{\exp(-y)}{(1 + \exp(-y))} \cdot \frac{1}{(1 + \exp(-y))} = 2\sigma(y)(1 - \sigma(y)) \quad (3.3)$$

Где $\sigma(y) = \frac{1}{1 + \exp(-y)} \in [0,1]$ - классическая функция сигмоиды

В конце численного решения данной задачи выполняется отображение вектора z_i на множество значений $\{1,-1\}$ по правилу

$$t_i = f(z_i) = \begin{cases} 1, & \text{если } z_i \geq z_{hop} \\ -1, & \text{если } z_i < z_{hop} \end{cases}, i = \overline{1, m} \quad (3.4)$$

В формуле (3.4) z_{hop} - пороговое значение, по умолчанию равно нулю.

В данной задаче используется модель обучения нейронной сети "с учителем". То есть, для всех состояний $N = 2^n$ входного сигнала вектор функции учителя $\overline{z_i^k}, i = \overline{1, m}, k = \overline{0, N-1}$ показывает, каким должен быть безошибочным сигнал на выходе нейронной сети.

Таким образом, функция ошибок суммирует четвертую степень разности функции учителя и выходного сигнала для каждой компоненты выходного вектора и для каждого возможного вектора входного сигнала. В случае если число входных состояний сигнала меньше даже числа переменных весовых коэффициентов в задаче $N = 2^n < m \times (n+1)$, то нужно взять повторяющийся входной сигнал с числом состояний кратному основному $N = p \cdot 2^n, p \in N$. С учетом оптимизации функции ошибок итерационным методом градиентного спуска также кратно числу различных состояний $N = 2^n$ входного сигнала приходится увеличивать число итераций процесса обучения нейронной сети.

На выходе нейронной сети умножим вектор обработанного сигнала на квадратную поворотную матрицу $D_{i,s}, i, s = \overline{1, m}$, увеличивая число весовых переменных коэффициентов еще на m^2 свободных переменных. Имеем

$$z_i^k = \sum_{s=1}^m D_{i,s} \overline{\sigma \left(\sum_{j=1}^n C_{s,j} x_j + b_s \right)}, i = \overline{1, m}, k = \overline{0, N-1} = \overline{0, 2^n - 1} \quad (3.5)$$

В итоге получим функцию ошибок

$$E(C_{s,j}, b_s, D_{i,s}) = \sum_{k=0}^{N-1} \sum_{i=1}^m \left(z_i^k - \overline{z_i^k} \right)^4 = \sum_{k=0}^{N-1} \sum_{i=1}^m \left(\sum_{s=1}^m D_{i,s} \overline{\sigma \left(\sum_{j=1}^n C_{s,j} x_j + b_s \right)} - \overline{z_i^k} \right)^4 \quad (3.6)$$

Обратим внимание на то, что функцию принятия решения по формуле (3.4) мы применяем один раз после того как по циклу численно решена задача поиска минимума функции ошибок (3.6) методом градиентного спуска. Внутри итерационного цикла переменная z_i^k меняется, оставаясь действительным числом, а в конце алгоритма каждое значение $z_i^{N-1}, i = \overline{1, m}$ скачком отображается в число -1 либо в 1. Для того чтобы величина градиента была достаточно большой и уменьшить число итераций в цикле алгоритма градиентного спуска суммируются не квадраты невязок задачи, а четвертые степени невязок.

Найдем компоненты градиента функции ошибок

$$\begin{aligned} \frac{\partial E(C_{s,j}, b_s, D_{i,s})}{\partial C_{s,j}} &= \sum_{k=0}^{N-1} \sum_{i=1}^m 4 \left(\sum_{s=1}^m D_{i,s} \overline{\sigma \left(\sum_{j=1}^n C_{s,j} x_j + b_s \right)} - \overline{z_i^k} \right)^3 D_{i,s} 2\sigma(y_s)(1-\sigma(y_s))x_j = \\ &= 8 \sum_{k=0}^{N-1} \sum_{i=1}^m \left(\sum_{s=1}^m D_{i,s} \overline{\sigma \left(\sum_{j=1}^n C_{s,j} x_j + b_s \right)} - \overline{z_i^k} \right)^3 D_{i,s} \sigma(y_s)(1-\sigma(y_s))x_j, y_s = \sum_{j=1}^n C_{s,j} x_j + b_s, s = \overline{1, m}, j = \overline{1, n} \quad (3.7) \end{aligned}$$

Замечание 3.1. Именно благодаря тому, что в формуле (3.7) $x_j = \{-1, 1\}$, $\sigma(y_s) > 0, 1 - \sigma(y_s) > 0$ $\frac{\partial E(C_{s,j}, b_s, D_{i,s})}{\partial C_{s,j}} = 0 \Leftrightarrow \sum_{s=1}^m D_{i,s} \bar{\sigma} \left(\sum_{j=1}^n C_{s,j} x_j + b_s \right) - \bar{z}_i^{-k} = 0$, что равносильно тому, что во всех случаях выходной сигнал совпадает с функцией учителя. Иначе, если $x_j = 0$, мы получим равенство нулю компоненты градиента даже в случае, когда выходной сигнал не совпадает со значением функции учителя в точке $x_j = 0$. То есть, имеем "фиктивные", не настоящие точки локального минимума функции ошибок (3.6).

Аналогично получим

$$\begin{aligned} \frac{\partial E(C_{s,j}, b_s, D_{i,s})}{\partial b_s} &= \sum_{k=0}^{N-1} \sum_{i=1}^m 4 \left(\sum_{s=1}^m D_{i,s} \bar{\sigma} \left(\sum_{j=1}^n C_{s,j} x_j + b_s \right) - \bar{z}_i^{-k} \right)^3 D_{i,s} 2\sigma(y_s)(1 - \sigma(y_s)) x_j = \\ &= 8 \sum_{k=0}^{N-1} \sum_{i=1}^m \left(\sum_{s=1}^m D_{i,s} \bar{\sigma} \left(\sum_{j=1}^n C_{s,j} x_j + b_s \right) - \bar{z}_i^{-k} \right)^3 \sigma(y_s)(1 - \sigma(y_s)) D_{i,s}, s = \overline{1, m} \end{aligned} \quad (3.8)$$

$$\frac{\partial E(C_{s,j}, b_s, D_{i,s})}{\partial D_{i,s}} = \sum_{k=0}^{N-1} 4 \left(\sum_{s=1}^m D_{i,s} \bar{\sigma} \left(\sum_{j=1}^n C_{s,j} x_j + b_s \right) - \bar{z}_i^{-k} \right)^3 D_{i,s} \bar{\sigma}(y_s), i, s = \overline{1, m} \quad (3.9)$$

Сравнение формул (3.6) и (3.9) показывает, что в формуле (3.9) нет суммирования по индексам i и частично по индексу s (в конце(3.9)), так как данные индексы содержит коэффициент $D_{i,s}$.

В итоге получим формулы для алгоритма градиентного спуска

$$\begin{cases} C_{s,j}^{l+1} = C_{s,j}^l - h_1 \frac{\partial E(C_{s,j}^l, b_s^l, D_{i,s}^l)}{\partial C_{s,j}}, l = 1, 2, \dots, N_1, N_1 = p2^n, s = \overline{1, m}, j = \overline{1, n} \\ b_s^{l+1} = b_s^l - h_1 \frac{\partial E(C_{s,j}^l, b_s^l, D_{i,s}^l)}{\partial b_s} \\ D_{i,s} = D_{i,s} - h_2 \frac{\partial E(C_{s,j}^l, b_s^l, D_{i,s}^l)}{\partial D_{i,s}}, i, s = \overline{1, m} \end{cases} \quad (3.10)$$

В формулах (3.10) h_1, h_2 – 2 различных шага градиентного спуска, а l – номер итерации

Замечание 3.2. Отметим, что в формулах градиентного спуска (3.10) два разных шага h_1, h_2 , так как в классической многослойной нейронной сети минимизацию функции ошибок иногда проводят последовательно от слоя к слою нейронов, то есть оптимизируют только те весовые коэффициенты, от которых зависит только данный промежуток между слоями, а, следовательно, можно использовать различные шаги градиентного спуска до нейронов и после нейронов в модели мономерной нейронной сети. Практический анализ показывает диапазон связи шагов $h_2 = (h_1/3) - (h_1/10)$.

Замечание 3.3. Можно построить мягкую модифицированную сигмоиду последовательно применяя композицию двух сигмoids.

$$z_s = \bar{\bar{\sigma}}(y_s), s = \overline{1, m}, \bar{\bar{\sigma}}(y) = \bar{\sigma}(\bar{\sigma}(y)) = \frac{2}{1 + \exp\left(-\left(\frac{2}{1 + \exp(-y)} - 1\right)\right)} - 1 \in [-0,46212; 0,46212], y \in R \quad (3.11)$$

$$\overline{\sigma'}(y) = \overline{\sigma}(\overline{\sigma}(y)) \overline{\sigma'}(y) = 2\sigma(\overline{\sigma}(y)) \cdot (1 - \sigma(\overline{\sigma}(y))) \cdot 2\sigma(y)(1 - \sigma(y)) = 4\sigma(\overline{\sigma}(y)) \cdot (1 - \sigma(\overline{\sigma}(y))) \cdot \sigma(y)(1 - \sigma(y))$$

Используем формулы алгоритма (3.6), (3.7), (3.8), (3.9), (3.10) для программы на языке Fortran, выбор языка объясняется близким математическим написанием формул, как в тексте работы, так и в коде программы, кроме того, умножение матриц сводится к упрощенной записи $C(:, :) = \text{matmull}(A(:, :), B(:, :))$ и экономии числа вложенных циклов. Использование мягкой композиции 2 модифицированных сигмоид делает нечувствительными результаты работы программы, если в формуле градиента (3.8) заменить местами аргумент y_s и значение функции $\sigma(y_s)$.

```
! 2 storonnya mono sets, z=g(g(x)) 2 matrici y=Cx+b z=Dg(y)
program net
use dfimsl
use msimsl
use msflib
integer(8), parameter:: m=4,n=4,n1=15,n0=2, n2=176 !n2=(n1+1)*n0
integer(8)::i,j,k,kk,p,xx,yy,x(n),teacher(n,0:n1)
real(8)::cc1(m,n),cc2(m,n),b1(m),b2(m),y(m),x1,g,gg,gg1,porog,x2,s,s1,s2,s3,s4
real(8)::h,e0,mas2(n,0:n1),teacher2(m,0:n1),mas(m,0:n1),z(m),eps,ec(m,n),eb(m)
real(8):: status2,xe,ye,m1(0:n1),m2(0:n1),m3(0:n1),res0(m,0:n1),d1(m,m),d2(m,m),ed(m,m)
real(8)::z1(m),mas0(m,0:n1),x0,z2(m),z3(m),gg0,gg2,gg3
type(windowconfig) wc
data wc.numxpixels,wc.numypixels,wc.numtextcols, wc.numtextrows, wc.numcolors,
wc.fontsize/6*-1/
g(x1)=1d0/(1d0+dexp(-x1))
gg(x1)=2d0/(1d0+dexp(-x1))-1d0
gg1(x1)=2d0*g(x1)*(1d0-g(x1))
gg0(x1)=gg(gg(x1))
gg2(x1)=gg1(gg(x1))*gg1(x1)
gg3(x1)=2d0/(1d0+1d0/dexp(gg(x1)))-1d0
res=setwindowconfig(wc)
res=getwindowconfig(wc)
xe=wc.numxpixels
ye=wc.numypixels
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
print*, " one neuron net"
h1=8d-3;h2=h1/9d0;porog=0d0
do i=1,m
do j=1,n
cc1(i,j)=0d0;cc2(i,j)=0d0;
enddo
enddo
ed(:,:)=0d0;
do i=1,m
b1(i)=0d0;b2(i)=0d0;
enddo

do i=1,m
do j=1,m
if(i==j)then
d1(i,j)=1d0
d2(i,j)=1d0
```

```

else
d1(i,j)=0d0
d2(i,j)=0d0
endif
enddo
enddo
do k=0,n1
xx=k;
do i=1,n
yy=mod(xx,2);x(i)=yy
xx=(xx-yy)/2;
teacher(i,k)=2*x(i)-1
mas2(i,k)=dfloat(2*x(i)-1)
enddo
enddo
do k=0,n1
do i=1,m
teacher2(i,k)=mas2(i,n1-int(k/3))
enddo
enddo
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
do p=1,n2

do k=0,n1

y(:)=matmul(cc1(:,:),mas2(:,k))+b1(:)
z1(:)=y(:)
do i=1,n0
do j=1,m
z2(j)=gg(z1(j))
enddo
z1(:)=z2(:)
enddo
z(:)=z1(:)
do i1=1,m
mas0(i1,k)=z(i1)
enddo
z1(:)=matmul(d1(:,:),z(:))
do i1=1,m
mas(i1,k)=z1(i1)
enddo
enddo

do kk=1,m
do j=1,n
s=0d0
s2=0d0;
do i=1,m
do k=0,n1
x2=mas2(j,k)
x1=mas(i,k)
s=s+d1(i,kk)*(4d0*gg1(x1)*(x1-teacher2(i,k))**3d0)*x2

```

```

s2=s2+d1(i,kk)*4d0*gg1(x1)*(x1-teacher2(i,k))**3d0
enddo
enddo
ec(kk,j)=s
!ec(kk,j)=s*dfloat(m)
eb(kk)=s2
!eb(kk)=s2*dfloat(m)
enddo
enddo

do i=1,m
do kk=1,m
s1=0d0
do k=0,n1
x1=mas(i,k)
s1=s1+(4d0*(x1-teacher2(i,k))**3d0)*mas0(kk,k)
enddo
ed(i,kk)=s1
enddo
enddo

do i=1,m
b2(i)=b1(i)-h1*eb(i)
enddo
do i=1,m
b1(i)=b2(i)
enddo

do i=1,m
do j=1,n
cc2(i,j)=cc1(i,j)-h1*ec(i,j)
enddo
enddo
do i=1,m
do j=1,n
cc1(i,j)=cc2(i,j)
enddo
enddo

do i=1,m
do j=1,m
d2(i,j)=d1(i,j)-h2*ed(i,j)
enddo
enddo
do i=1,m
do j=1,m
d1(i,j)=d2(i,j)
enddo
enddo
enddo
do k=0,n1
y(:)=matmul(cc1(:,k),mas2(:,k))+b1(:)

```

```

do i=1,m
s=gg(y(i))
z(i)=s
enddo

z1(:)=matmul(d1(:,z),z(:))

do i=1,m
if(z1(i)<porog)then
z1(i)=-1d0
elseif(z1(i)>=porog)then
z1(i)=1d0
endif
res0(i,k)=z1(i)
enddo
enddo
!!!!!!!!!!!!!!!!!!!!funkcia oshibok
eps=0d0;s1=0d0
do k=0,n1
s=0d0
do i=1,m
s=s+(res0(i,k)-teacher2(i,k))*4d0
s1=s1+s
enddo
if(s<=1d-1)then
eps=eps
elseif(s>1d-1)then
eps=eps+1d0
endif
enddo
e0=s1
print*, "e0=",e0,"eps=",eps
do i=0,15
m1(i)=dfloat(i)
s1=0d0
do j=1,m
if(teacher2(j,i)<0d0)then
s=0d0
elseif(teacher2(j,i)>0d0)then
s=1d0
endif
s1=s1+s*2d0**dfloat(j-1)
enddo
m2(i)=s1
s3=0d0
do j=1,m
if(res0(j,i)<0d0)then
s4=0d0
elseif(res0(j,i)>0d0)then
s4=1d0
endif
s3=s3+s4*2d0**dfloat(j-1)

```

```

enddo
m3(i)=s3
enddo
call axis1(m1)
pause
call axis2(m2)
pause
call axis3(m3)
pause
!!!!!!!!!!!!!!!!!!!!!!
contains

```

```

subroutine axis1(m1)
type(xycoord) xy
integer(8)::i,j
real(8)::status2,m1(0:n1)
do i=0,n1
status2=setcolor(m1(i)+0_8)
do j=0,10
call moveto(int2(xe/10d0)-10,int2((ye-dfloat(15*i+j))/2),xy)
status2=lineto(1_1*xe/10d0+10_1,(ye-dfloat(15*i+j))/2_2)
enddo
enddo
end subroutine axis1

```

```

subroutine axis2(m2)
type(xycoord) xy
integer(8)::i,j
real(8)::status2,m2(0:n1)
do i=0,n1
status2=setcolor(m2(i)+0_8)
do j=0,10
call moveto(int2((xe)/50)-10,int2((ye-dfloat(15*i+j))/2),xy)
status2=lineto(1_1*(xe)/50+10_1,(ye-dfloat(15*i+j))/2_2)
enddo
enddo
end subroutine axis2

```

```

subroutine axis3(m3)
type(xycoord) xy
integer(8)::i
real(8)::status2,m3(0:n1)
do i=0,n1
status2=setcolor(m3(i)+0_8)
do j=0,10
call moveto(int2((xe)/5)-10,int2((ye-dfloat(15*i+j))/2),xy)
status2=lineto(1_1*(xe)/5+10_1,(ye-dfloat(15*i+j))/2_2)
enddo
enddo
end subroutine axis3
end

```

Для тестирования второй программы и сравнения результатов ее работы с результатами работы первой программы используем по-прежнему задачу преобразования цветности предмета, а также задачу сжатия информации о цветности. В программе $m=n=4$, число итераций в цикле равно $n_2=176$ (кратно 16, $p=11$), номер цвета меняется от 0 (черный цвет) до 15 (белый цвет). Кроме того, первые цвета с номерами от 0 до 7 являются тусклыми, а от 8 до 15 яркими цветами. Для функции активации использована формула (3.11) мягкой модифицированной сигмоиды. Рассмотрим формулу преобразования цветности

$$n_2(i) = N_1 - 1 - \text{mod}(i, n_3) \quad (3.12)$$

- 1) Пример, в центральном столбце исходная последовательность цвета на входе нейронной сети, слева функция учителя с инверсией цвета, справа обученная нейронная сеть

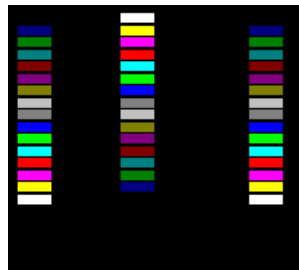


Рис.1. Случай инверсии цветности $n_2(i) = 15 - \text{mod}(i,1) = 15 - i$
 $\epsilon_0 = 0.0000000000000000E+000$ $\text{eps} = 0.0000000000000000E+000$ - программа дает функцию ошибок и число неправильно обученных цветов равных нулю.

- 2) Пример - задача о сжатии цвета в 2 раза: $n_2(i) = 15 - \text{mod}(i,2)$

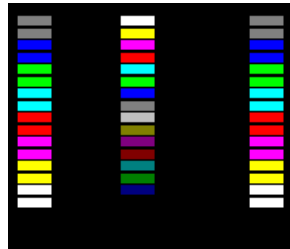


Рис.2. Задача о сжатии цветности в 2 раза. $\epsilon_0 = 0.0000000000000000E+000$ $\text{eps} = 0.0000000000000000E+000$ - программа дает функцию ошибок и число неправильно обученных цветов равных нулю.

- 3) Пример- задача о сжатии цвета в 3 раза: $n_2(i) = 15 - \text{mod}(i,3)$

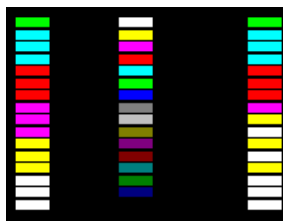


Рис.3. Задача о сжатии цветности в 3 раза. $\epsilon_0 = 224.00000000000000$ $\text{eps} = 3.0000000000000000$ - программа дает функцию ошибок равной 224 и 3 неправильно обученных цвета.

- 4) Пример - задача о сжатии цвета в 4 раза: $n_2(i) = 15 - \text{mod}(i,4)$

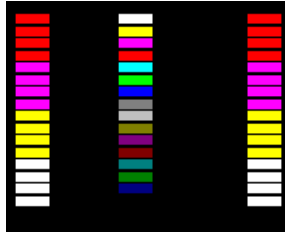


Рис.4. Задача о сжатии цветности в 4 раза. $\epsilon_0 = 0.000000000000$ $\epsilon_{ps} = 0.0000000000000000$ - программа дает функцию ошибок равной 0 и 0 неправильно обученных цвета.

5) Пример - задача о сжатии цвета в 5 раз: $n_2(i) = 15 - \text{mod}(i,5)$

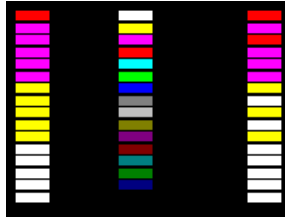


Рис.5. Задача о сжатии цветности в 5 раз. $\epsilon_0 = 192.000000000000$ $\epsilon_{ps} = 3.0000000000000000$ - программа дает функцию ошибок равной 192 и 3 неправильно обученных цвета.

6) Пример - задача о сжатии цвета в 6 раз: $n_2(i) = 15 - \text{mod}(i,6)$

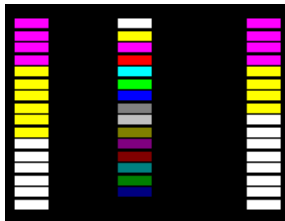


Рис.6. Задача о сжатии цветности в 6 раз. $\epsilon_0 = 128.000000000000$ $\epsilon_{ps} = 2.0000000000000000$ - программа дает функцию ошибок равной 128 и 2 неправильно обученных цвета.

7) Пример - задача о сжатии цвета в 7 раз: $n_2(i) = 15 - \text{mod}(i,7)$

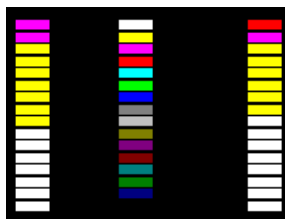


Рис.7. Задача о сжатии цветности в 7 раз. $\epsilon_0 = 128.000000000000$ $\epsilon_{ps} = 2.0000000000000000$ - программа дает функцию ошибок равной 128 и 2 неправильно обученных цвета.

8) Пример - задача о сжатии цвета в 8 раз: $n_2(i) = 15 - \text{mod}(i,8)$

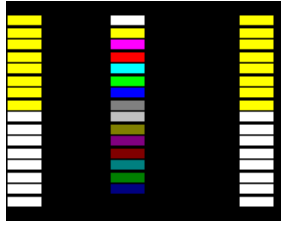


Рис.8. Задача о сжатии цветности в 8 раз. $e_0 = 0.000000000000$ $\epsilon = 0.000000000000$ - программа дает функцию ошибок равной 0 и 0 неправильно обученных цвета.

9) Пример - задача о сжатии цвета в 9 раз: $n_2(i) = 15 - \text{mod}(i,9)$

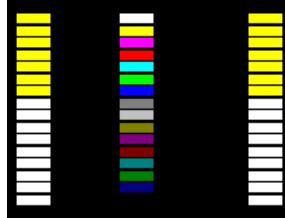


Рис.9. Задача о сжатии цветности в 9 раз. $e_0 = 0.000000000000$ $\epsilon = 0.000000000000$ - программа дает функцию ошибок равной 0 и 0 неправильно обученных цвета.

10) Пример - задача о сжатии цвета в 10 раз: $n_2(i) = 15 - \text{mod}(i,10)$

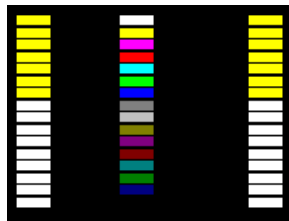


Рис.10. Задача о сжатии цветности в 10 раз. $e_0 = 0.000000000000$ $\epsilon = 0.000000000000$ - программа дает функцию ошибок равной 0 и 0 неправильно обученных цвета.

Замечание 4.2. Анализируя все случаи стопроцентного обучения нейронной сети можно сказать, что они реализуются, когда число состояний входного сигнала (16) делятся нацело на модуль (1,2,4,8), а также когда модуль больше 8.

Замечание 5.2. Из всех случаев обучения нейронной моносети необходимо выбрать худший (13/16=81,25% обучения- 3 неправильных цвета из 16) и взвешенный процент обучения, вычисленный по формуле

$$\bar{E} = \frac{1}{N} \sum_{i=0}^{N-1} n_i p_i = \frac{1}{16} (12 \cdot 1 + 2 \cdot 13/16 + 2 \cdot 14/16) = 0,9609 \approx 96,1\%$$

Лекция 4. Задача третья. Прямоугольная матрица плюс вектор весовых коэффициентов на входе и квазиединичная квадратная матрица на выходе нейронного монослоя. Практическая работа 9.

Данная математическая модель существенно упрощает вторую задачу. Вектор входного сигнала представлен в двоичном коде и принимает всего два значения $\{1,-1\}$ и имеет размерность n . Данное множество из двух чисел является аналогом множества остатков целых чисел по модулю два $\{1,0\}$. Вектор сигнала, принимаемый нейронами, имеет другую размерность m и равен

$$y = Cx + b, \quad y_s = \sum_{j=1}^n C_{s,j} x_j + b_s, \quad s = \overline{1,m}, \quad x_j = \{-1,1\}, \quad j = \overline{1,n} \quad (4.1)$$

В формуле (4.1) $C_{s,j}, s = \overline{1,m}, j = \overline{1,n}$ – прямоугольная матрица весовых коэффициентов, $b_s, s = \overline{1,m}$ – вектор весовых коэффициентов. В пространстве размерности $m \times (n+1)$ изменением матрицы $C_{s,j}, s = \overline{1,m}, j = \overline{1,n}$ и вектора $b_s, s = \overline{1,m}$ проводится оптимизация задачи, то есть минимизируется функционал ошибки.

На выходе нейронов сигнал отличается от входного сигнала тем, что он обрабатывается функцией активации. Для простоты мы выбрали наиболее распространенную и универсальную, но модифицированную функцию “сигмоиды”:

$$z_s = \overline{\sigma}(y_s), s = \overline{1,m}, \overline{\sigma}(y) = \frac{2}{1 + \exp(-y)} - 1 \in [-1,1], y \in R \quad (4.2)$$

$$\overline{\sigma}'(y) = \frac{d\overline{\sigma}(y)}{dy} = \frac{2 \exp(-y)}{(1 + \exp(-y))^2} = 2 \frac{\exp(-y)}{(1 + \exp(-y))} \cdot \frac{1}{(1 + \exp(-y))} = 2\sigma(y)(1 - \sigma(y)) \quad (4.3)$$

Где $\sigma(y) = \frac{1}{1 + \exp(-y)} \in [0,1]$ - классическая функция сигмоиды

После численного решения задачи алгоритмом градиентного спуска выполняется отображение вектора z_i на множество значений $\{1, -1\}$ по правилу

$$t_i = f(z_i) = \begin{cases} 1, & \text{если } z_i \geq z_{hop} \\ -1, & \text{если } z_i < z_{hop} \end{cases}, i = \overline{1,m} \quad (4.3)$$

В формуле (4.3) z_{hop} - пороговое значение, по умолчанию равно нулю.

В данной задаче используется упрощенная модель обучения нейронной сети ” с учителем”. То есть, для всех состояний $N = 2^n$ входного сигнала вектор функции учителя $\overline{z}_i^k, i = \overline{1,m}, k = \overline{0, N-1}$ показывает, каким должен быть безошибочным сигнал на выходе нейронной сети.

Таким образом, функция ошибок суммирует четвертую степень разности функции учителя и выходного сигнала для каждой компоненты выходного вектора и для каждого возможного вектора входного сигнала. В случае если число входных состояний сигнала меньше даже числа переменных весовых коэффициентов в задаче $N = 2^n < m \times (n+1)$, то нужно взять повторяющийся входной сигнал с числом состояний кратному основному $N = p \cdot 2^n, p \in N$. С учетом оптимизации функции ошибок итерационным методом градиентного спуска также кратно числу различных состояний $N = 2^n$ входного сигнала приходится увеличивать число итераций процесса обучения нейронной сети.

На выходе нейронной сети умножим вектор обработанного сигнала на квадратную поворотную матрицу $D_{i,s}, i, s = \overline{1,m}$, увеличивая число весовых переменных коэффициентов еще на m^2 свободных переменных. сначала $D_{i,s} = \delta_{i,s} = \{1 \Leftrightarrow i = s, 0 \Leftrightarrow i \neq s\}$

$$z_i^k = \sum_{s=1}^m D_{i,s} \overline{\sigma} \left(\sum_{j=1}^n C_{s,j} x_j^k + b_s \right) = \overline{\sigma} \left(\sum_{j=1}^n C_{s,j} x_j^k + b_s \right), k = \overline{0, N-1}, s = \overline{1,m} \quad (4.4)$$

В итоге получим функцию ошибок

$$E(C_{s,j}, b_s, D_{i,s}) = \sum_{k=0}^{N-1} \sum_{i=1}^m \left(z_i^k - \overline{z}_i^k \right)^4 = \sum_{k=0}^{N-1} \sum_{i=1}^m \left(\overline{\sigma} \left(\sum_{j=1}^n C_{s,j} x_j^k + b_s \right) - \overline{z}_i^k \right)^4 \quad (4.5)$$

Для того чтобы величина градиента была достаточно большой и уменьшить число итераций в цикле алгоритма градиентного спуска суммируются не квадраты невязок задачи, а четвертые степени невязок.

Найдем компоненты градиента функции ошибок

$$\begin{aligned} \frac{\partial E(C_{s,j}, b_s, D_{i,s})}{\partial C_{s,j}} &= \sum_{k=0}^{N-1} \sum_{i=1}^m 4 \left(\bar{\sigma} \left(\sum_{j=1}^n C_{s,j} x_j + b_s \right) - \bar{z}_s^{-k} \right)^3 2\sigma(y_s)(1-\sigma(y_s))x_j = \\ &= 8m \sum_{k=0}^{N-1} \left(\bar{\sigma} \left(\sum_{j=1}^n C_{s,j} x_j + b_s \right) - \bar{z}_s^{-k} \right)^3 \sigma(y_s)(1-\sigma(y_s))x_j, y_s = \sum_{j=1}^n C_{s,j} x_j + b_s, s = \overline{1, m}, j = \overline{1, n} \end{aligned} \quad (4.6)$$

Аналогично получим

$$\begin{aligned} \frac{\partial E(C_{s,j}, b_s, D_{i,s})}{\partial b_s} &= 8 \sum_{k=0}^{N-1} \sum_{i=1}^m \left(\bar{\sigma} \left(\sum_{j=1}^n C_{s,j} x_j + b_s \right) - \bar{z}_s^{-k} \right)^3 \sigma(y_s)(1-\sigma(y_s)) = \\ &= 8m \sum_{k=0}^{N-1} \left(\bar{\sigma} \left(\sum_{j=1}^n C_{s,j} x_j + b_s \right) - \bar{z}_s^{-k} \right)^3 \sigma(y_s)(1-\sigma(y_s)) \end{aligned} \quad (4.7)$$

$$\frac{\partial E(C_{s,j}, b_s, D_{i,s})}{\partial D_{i,s}} = \sum_{k=0}^{N-1} 4 \left(\bar{\sigma} \left(\sum_{j=1}^n C_{s,j} x_j + b_s \right) - \bar{z}_s^{-k} \right)^3 \bar{\sigma}(y_s), i, s = \overline{1, m} \quad (4.8)$$

В формулах (4.6), (4.7), (4.8) нет суммирования по индексам i , так как в (4.6), (4.7) выражение под знаком суммы не зависит от i , а в (4.8) данные индексы i, s содержит коэффициент $D_{i,s}$.

В итоге получим формулы для алгоритма градиентного спуска

$$\begin{cases} C_{s,j}^{l+1} = C_{s,j}^l - h_1 \frac{\partial E(C_{s,j}^l, b_s^l, D_{i,s}^l)}{\partial C_{s,j}}, l = 1, 2, \dots, N_1, N_1 = p2^n, s = \overline{1, m}, j = \overline{1, n} \\ b_s^{l+1} = b_s^l - h_1 \frac{\partial E(C_{s,j}^l, b_s^l, D_{i,s}^l)}{\partial b_s} \\ D_{i,s} = D_{i,s} - h_2 \frac{\partial E(C_{s,j}^l, b_s^l, D_{i,s}^l)}{\partial D_{i,s}}, i, s = \overline{1, m} \end{cases} \quad (4.9)$$

В формулах (4.9) h_1, h_2 – различных шага градиентного спуска, а l – номер итерации

Замечание 4.1. Отметим, что в формулах градиентного спуска (4.9) два разных шага h_1, h_2 , так как в классической многослойной нейронной сети минимизацию функции ошибок иногда проводят последовательно от слоя к слою нейронов, то есть оптимизируют только те весовые коэффициенты, от которых зависит только данный промежуток между слоями, а, следовательно, можно использовать различные шаги градиентного спуска до нейронов и после нейронов в модели мономерной нейронной сети. Практический анализ показывает диапазон связи шагов $h_2 = (h_1/3) - (h_1/10)$.

Используем формулы алгоритма (4.6), (4.7), (4.8), (4.9), (4.5) для программы на языке Fortran, выбор языка объясняется близким математическим написанием формул, как в тексте работы, так и в коде программы, кроме того, умножение матриц сводится к упрощенной записи $C(:, :) = \text{matmull}(A(:, :), B(:, :))$ и экономии числа вложенных циклов. Использование мягкой композиции 2 модифицированных сигмоид делает нечувствительными результаты работы программы, если в формуле градиента (4.7) заменить местами аргумент y_s и значение функции $\sigma(y_s)$.

```

! 2 storonnya mono sets, z=g(g(x))
program net
use dfimsl
use msimsl
use msflib
integer(8), parameter:: m=4,n=4,n1=15,n0=2, n2=96 !n2=(n1+1)*n0
integer(8)::i,j,k,kk,p,xx,yy,x(n),teacher(n,0:n1)
real(8)::cc1(m,n),cc2(m,n),b1(m),b2(m),y(m),x1,g,gg,gg1,porog,x2,s,s1,s2,s3,s4
real(8)::h,e0,mass2(n,0:n1),teacher2(m,0:n1),mas(m,0:n1),z(m),eps,ec(m,n),eb(m)
real(8):: status2,xe,ye,m1(0:n1),m2(0:n1),m3(0:n1),res0(m,0:n1),d1(m,m),d2(m,m),ed(m,m)
real(8)::z1(m),mas0(m,0:n1),x0,z2(m),z3(m),gg0,gg2,gg3
type(windowconfig) wc
data wc.numxpixels,wc.numypixels,wc.numtextcols, wc.numtextrows, wc.numcolors,
wc.fontSize/6*-1/
g(x1)=1d0/(1d0+dexp(-x1))
gg(x1)=2d0/(1d0+dexp(-x1))-1d0
gg1(x1)=2d0*g(x1)*(1d0-g(x1))
gg0(x1)=gg(gg(x1))
gg2(x1)=gg1(gg(x1))*gg1(x1)
gg3(x1)=2d0/(1d0+1d0/dexp(gg(x1)))-1d0
res=setwindowconfig(wc)
res=getwindowconfig(wc)
xe=wc.numxpixels
ye=wc.numypixels
h1=5d-3;h2=h1/3d0;porog=0d0
print*,x1,g(x1),gg(x1),gg1(x1)
do i=1,m
do j=1,n
cc1(i,j)=0d0;cc2(i,j)=0d0;
enddo
enddo
ed(:,:)=0d0;
do i=1,m
b1(i)=0d0;b2(i)=0d0;
enddo

do i=1,m
do j=1,m
if(i==j)then
d1(i,j)=1d0
d2(i,j)=1d0
else
d1(i,j)=0d0
d2(i,j)=0d0
endif
enddo
enddo

do k=0,n1
xx=k;
do i=1,n
yy=mod(xx,2);x(i)=yy

```

```

xx=(xx-yy)/2;
teacher(i,k)=2*x(i)-1
mas2(i,k)=dfloat(2*x(i)-1)
enddo
enddo

do k=0,n1
do i=1,m
teacher2(i,k)=mas2(i,n1-int(k/3))
enddo
enddo

do p=1,n2
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
do k=0,n1

y(:)=matmul(cc1(:,:),mas2(:,k))+b1(:)
z1(:)=y(:)
do i=1,n0
do j=1,m
z2(j)=gg(z1(j))
enddo
z1(:)=z2(:)
enddo
z(:)=z1(:)
do i1=1,m
mas0(i1,k)=z(i1)
enddo
z1(:)=matmul(d1(:,:),z(:))
do i1=1,m
mas(i1,k)=z1(i1)
enddo
enddo

do kk=1,m
do j=1,n
s=0d0
s2=0d0;
!do i=1,m
do k=0,n1
x2=mas2(j,k)
x1=mas(kk,k)
s=s+(4d0*gg1(x1)*(x1-teacher2(kk,k))**3d0)*x2
s2=s2+4d0*gg1(x1)*(x1-teacher2(kk,k))**3d0
enddo
ec(kk,j)=s*dfloat(m)
eb(kk)=s2*dfloat(m)
enddo
enddo

do i=1,m
do kk=1,m

```

```

s1=0d0
do k=0,n1
x1=mas(i,k)
s1=s1+(4d0*(x1-teacher2(i,k))**3d0)*mas0(kk,k)
enddo
ed(i,kk)=s1
enddo
enddo

```

```

do i=1,m
b2(i)=b1(i)-h1*eb(i)
enddo
do i=1,m
b1(i)=b2(i)
enddo

```

```

do i=1,m
do j=1,n
cc2(i,j)=cc1(i,j)-h1*ec(i,j)
enddo
enddo
do i=1,m
do j=1,n
cc1(i,j)=cc2(i,j)
enddo
enddo

```

```

do i=1,m
do j=1,m
d2(i,j)=d1(i,j)-h2*ed(i,j)
enddo
enddo
do i=1,m
do j=1,m
d1(i,j)=d2(i,j)
enddo
enddo
enddo

```

```

!!!!!!!!!!!!!!!!!!!!!!proverka
do k=0,n1
y(:)=matmul(cc1(:,k),mas2(:,k))+b1(:)
do i=1,m
s=gg(y(i))
z(i)=s
enddo

```

```

z1(:)=matmul(d1(:,k),z(:))
do i=1,m
if(z1(i)<porog)then
z1(i)=-1d0
elseif(z1(i)>=porog)then
z1(i)=1d0

```

```

endif
res0(i,k)=z1(i)
enddo
enddo
!!!!!!!!!!!!!!!!!!!!funkcia oshibok
eps=0d0;s1=0d0
do k=0,n1
s=0d0
do i=1,m
s=s+(res0(i,k)-teacher2(i,k))**4d0
s1=s1+s
enddo
if(s<=1d-1)then
eps=eps
elseif(s>1d-1)then
eps=eps+1d0
endif
!pause
enddo
e0=s1
print*,"e0=",e0,"eps=",eps
pause
do i=0,15
m1(i)=dfloat(i)
s1=0d0
do j=1,m
if(teacher2(j,i)<0d0)then
s=0d0
elseif(teacher2(j,i)>0d0)then
s=1d0
endif
s1=s1+s*2d0**dfloat(j-1)
enddo
m2(i)=s1
s3=0d0
do j=1,m
if(res0(j,i)<0d0)then
s4=0d0
elseif(res0(j,i)>0d0)then
s4=1d0
endif
s3=s3+s4*2d0**dfloat(j-1)
enddo
m3(i)=s3
enddo
call axis1(m1)
pause
call axis2(m2)
pause
call axis3(m3)
pause
contains

```

```

subroutine axis1(m1)
type(xycoord) xy
integer(8)::i,j
real(8)::status2,m1(0:n1)
do i=0,n1
status2=setcolor(m1(i)+0_8)
do j=0,10
call moveto(int2(xe/10d0)-10,int2((ye-dfloat(15*i+j))/2),xy)
status2=lineto(1_1*xe/10d0+10_1,(ye-dfloat(15*i+j))/2_2)
enddo
enddo
end subroutine axis1

```

```

subroutine axis2(m2)
type(xycoord) xy
integer(8)::i,j
real(8)::status2,m2(0:n1)
do i=0,n1
status2=setcolor(m2(i)+0_8)
do j=0,10
call moveto(int2((xe)/50)-10,int2((ye-dfloat(15*i+j))/2),xy)
status2=lineto(1_1*(xe)/50+10_1,(ye-dfloat(15*i+j))/2_2)
enddo
enddo
end subroutine axis2

```

```

subroutine axis3(m3)
type(xycoord) xy
integer(8)::i
real(8)::status2,m3(0:n1)
do i=0,n1
status2=setcolor(m3(i)+0_8)
do j=0,10
call moveto(int2((xe)/5)-10,int2((ye-dfloat(15*i+j))/2),xy)
status2=lineto(1_1*(xe)/5+10_1,(ye-dfloat(15*i+j))/2_2)
enddo
enddo
end subroutine axis3
end

```

Для тестирования третьей программы и сравнения результатов ее работы с результатами работ первой и второй программ используем по-прежнему задачу преобразования цветности предмета, а также задачу сжатия информации о цветности. В программе $m=n=4$, число итераций в цикле равно $n_2=176$ (кратно 16, $p=11$), номер цвета меняется от 0 (черный цвет) до 15 (белый цвет). Кроме того, первые цвета с номерами от 0 до 7 являются тусклыми, а от 8 до 15 яркими цветами. Для функции активации использована формула (11.2) мягкой модифицированной сигмоиды. Рассмотрим формулу преобразования цветности

$$n_2(i) = N_1 - 1 - \text{mod}(i, n_3) \quad (4.10)$$

Все 16 примеров из задачи 3 повторяют рисунки и результаты из задачи 2 и поэтому здесь не приводятся, однако в программе для вычисления компонентов градиента по формулам (4.6), (4.7) используют на 1 цикл меньше, что ускоряет работу программы.

Глава 4. Дискретное преобразование Фурье.

Лекция 5. Одномерное дискретное преобразование Фурье.

Рассмотрим периодическую функцию, зависящую от времени, период которой равен T . Предположим, что за период T получено N экспериментальных данных с равномерным временным шагом $x_n = T \frac{n}{N}, n = \overline{0, N-1}$. Обозначим частоты периодической функции, минимальное значение имеет основная частота $\nu_k = \frac{k}{T}, k = \overline{0, N-1}$. Круговые частоты отличаются от частот ν_k множителем 2π : $\omega_k = \frac{2\pi k}{T}$. Разложим периодическую функцию $x(t)$ в ряд Фурье.

$x(t) = \sum_{k=-\infty}^{\infty} c_k e^{i\omega_k t}, i = \sqrt{-1}$. Тогда в дискретные моменты времени получим $x(t_n) = x_n = \sum_{k=-\infty}^{\infty} c_k e^{i\omega_k t_n} = \sum_{k=-\infty}^{\infty} c_k e^{i \frac{2\pi k}{T} T \frac{n}{N}} = \sum_{k=-\infty}^{\infty} c_k e^{i \frac{2\pi k n}{N}} = \sum_{k=0}^{N-1} X_k e^{i \frac{2\pi k n}{N}}, X_k = \sum_{l=-\infty}^{\infty} c_{k+lN}, e^{i \frac{2\pi k n}{N}} = e^{i \frac{2\pi(k+lN)n}{N}}$

Таким образом, мы получили формулу, которая носит название обратного преобразования Фурье

$$x_n = \sum_{k=0}^{N-1} X_k e^{i \frac{2\pi k n}{N}}, n = \overline{0, N-1}, X_k = \sum_{l=-\infty}^{\infty} c_{k+lN} \quad (5.1)$$

Левую и правую части формулы (5.1) умножим скалярно на множитель $e^{-i \frac{2\pi m n}{N}}, m = \overline{0, N-1}$, получим

$$x_n e^{-i \frac{2\pi m n}{N}} = \sum_{k=0}^{N-1} X_k e^{i \frac{2\pi k n}{N}} e^{-i \frac{2\pi m n}{N}}, n = \overline{0, N-1}.$$

Последнее выражение просуммируем по индексу n , имеем

$$\sum_{n=0}^{N-1} x_n e^{-i \frac{2\pi m n}{N}} = \sum_{n=0}^{N-1} \sum_{k=0}^{N-1} X_k e^{i \frac{2\pi(k-m)n}{N}} = \sum_{k=0}^{N-1} X_k \sum_{n=0}^{N-1} e^{i \frac{2\pi(k-m)n}{N}}, n, m = \overline{0, N-1} \quad (5.2)$$

Отдельно найдем сумму ряда

$$A = \sum_{n=0}^{N-1} e^{i \frac{2\pi(k-m)n}{N}} = 1 + e^{i \frac{2\pi(k-m)}{N}} + e^{i \frac{4\pi(k-m)}{N}} + \dots + e^{i \frac{2\pi(N-1)(k-m)}{N}} =$$

$$1 + e^{i \frac{2\pi(k-m)}{N}} \left(1 + e^{i \frac{2\pi(k-m)}{N}} + e^{i \frac{4\pi(k-m)}{N}} + \dots + e^{i \frac{2\pi(N-2)(k-m)}{N}} + e^{i \frac{2\pi(N-1)(k-m)}{N}} - e^{i \frac{2\pi(N-1)(k-m)}{N}} \right) =$$

$$= A = 1 + e^{i \frac{2\pi(k-m)}{N}} \left(A - e^{i \frac{2\pi(N-1)(k-m)}{N}} \right) \Leftrightarrow A = \frac{1 - e^{i \frac{2\pi N(k-m)}{N}}}{1 - e^{i \frac{2\pi(k-m)}{N}}} = \frac{1 - e^{i 2\pi(k-m)}}{1 - e^{i \frac{2\pi(k-m)}{N}}} = \begin{cases} \frac{1-1}{e^{i \frac{2\pi(k-m)}{N}} - 1} = 0, k \neq m \\ 1 - e^{i \frac{2\pi(k-m)}{N}} \\ \approx \frac{1 - (1 - i 2\pi(k-m))}{1 - (1 - i \frac{2\pi}{N}(k-m))} = N, k = m \end{cases}.$$

В итоге получим

$$\sum_{n=0}^{N-1} x_n e^{-i\frac{2\pi mn}{N}} = \sum_{n=0}^{N-1} \sum_{k=0}^{N-1} X_k e^{i\frac{2\pi m(k-m)}{N}} = \sum_{k=0}^{N-1} X_k \frac{1 - e^{i2\pi(k-m)}}{1 - e^{i\frac{2\pi(k-m)}{N}}} = \sum_{m=0}^{N-1} X_m N \delta_{k,m} = NX_k, n, k, m = \overline{0, N-1}$$

$$X_k = \frac{1}{N} \sum_{n=0}^{N-1} x_n e^{-i\frac{2\pi(m-k)n}{N}} = \frac{1}{N} \sum_{n=0}^{N-1} x_n e^{-i\frac{2\pi kn}{N}}, k = \overline{0, N-1} \quad (5.3)$$

Формула (5.3) носит название прямого дискретного преобразование Фурье. Однако можно встретить более удобные на практике группы формул дискретного преобразования Фурье.

$$\begin{cases} X_k = \sum_{n=0}^{N-1} x_n e^{-i\frac{2\pi kn}{N}}, k = \overline{0, N-1} \\ x_n = \frac{1}{N} \sum_{k=0}^{N-1} X_k e^{i\frac{2\pi kn}{N}}, n = \overline{0, N-1}. \end{cases} \quad (5.4)$$

$$\begin{cases} X_k = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} x_n e^{-i\frac{2\pi kn}{N}}, k = \overline{0, N-1} \\ x_n = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} X_k e^{i\frac{2\pi kn}{N}}, n = \overline{0, N-1}. \end{cases} \quad (5.5)$$

Обратим внимание, что пары формул (5.1), (5.3), а также (5.4),(5.5) отличаются только множителями перед суммами, но композиция прямого и обратного преобразований дает один общий множитель $s \stackrel{(5.1)(5.3)}{=} 1 \cdot \frac{1}{N} \stackrel{(5.4)}{=} \frac{1}{N} \cdot 1 \stackrel{(5.5)}{=} \frac{1}{\sqrt{N}} \frac{1}{\sqrt{N}}$.

Практическая работа 10. Одномерное дискретное преобразование Фурье

Воспользуемся формулами (5.4) дискретного преобразования Фурье.

$$\begin{cases} X_k = \sum_{n=0}^{N-1} x_n e^{-i\frac{2\pi kn}{N}}, k = \overline{0, N-1} \\ x_n = \frac{1}{N} \sum_{k=0}^{N-1} X_k e^{i\frac{2\pi kn}{N}}, n = \overline{0, N-1}. \end{cases} \quad (5.4)$$

Формула (5.4) может использовать как действительный временной набор данных, так и комплексный вектор $x_n, n = \overline{0, N-1}$. Если x_n действительный вектор, то из формулы (5.4) получим

$$X_k = \sum_{n=0}^{N-1} x_n e^{-i\frac{2\pi kn}{N}} = \sum_{n=0}^{N-1} x_n \cos\left(\frac{2\pi kn}{N}\right) - ix_n \sin\left(\frac{2\pi kn}{N}\right) = \sum_{n=0}^{N-1} (A_n + iB_n), i = \sqrt{-1}, k = \overline{0, N-1} \quad (5.6)$$

То есть спектральные коэффициенты прямого преобразования (спектр) получим исходя из формулы(5.6)

$$X_k = A_k + iB_k = \sum_{n=0}^{N-1} \left(x_n \cos\left(\frac{2\pi kn}{N}\right) - ix_n \sin\left(\frac{2\pi kn}{N}\right) \right) \Leftrightarrow$$

$$A_k = \sum_{n=0}^{N-1} x_n \cos\left(\frac{2\pi kn}{N}\right), B_k = -\sum_{n=0}^{N-1} x_n \sin\left(\frac{2\pi kn}{N}\right), k = \overline{0, N-1} \quad (5.7)$$

Используя формулы обратного дискретного преобразования Фурье (5.4), получим формулы для коэффициентов

$$\begin{aligned} x_n = c_n + id_n &= \frac{1}{N} \sum_{k=0}^{N-1} X_k e^{i\frac{2\pi kn}{N}} = \frac{1}{N} \sum_{k=0}^{N-1} (A_k + iB_k) e^{i\frac{2\pi kn}{N}} = \frac{1}{N} \sum_{k=0}^{N-1} (A_k + iB_k) \left(\cos\left(\frac{2\pi kn}{N}\right) + i \sin\left(\frac{2\pi kn}{N}\right) \right) = \\ &= \frac{1}{N} \sum_{k=0}^{N-1} \left(A_k \cos\left(\frac{2\pi kn}{N}\right) - B_k \sin\left(\frac{2\pi kn}{N}\right) \right) + i \left(B_k \cos\left(\frac{2\pi kn}{N}\right) + A_k \sin\left(\frac{2\pi kn}{N}\right) \right), n = \overline{0, N-1} \Leftrightarrow \\ \Leftrightarrow c_n &= \sum_{k=0}^{N-1} \left(A_k \cos\left(\frac{2\pi kn}{N}\right) - B_k \sin\left(\frac{2\pi kn}{N}\right) \right), d_n = \sum_{k=0}^{N-1} \left(B_k \cos\left(\frac{2\pi kn}{N}\right) + A_k \sin\left(\frac{2\pi kn}{N}\right) \right), n = \overline{0, N-1} \quad (5.8) \end{aligned}$$

В итоге, коэффициенты прямого дискретного преобразования Фурье в программе запишем по формулам (5.7), а коэффициенты обратного дискретного преобразования Фурье программируем по формулам (5.8).

Опишем алгоритм работы программы на языке C++. Все функции и аргументы в программе имеют двойную точность.

Пример 1. Исходная функция $x_i = \sin(ih)$, $h = (b-a)/N$, $a = 0$, $b = 2\pi$, $i = \overline{0, N-1}$, $N = 10$ является нечетной

```
//furie not chetnih function
#include<stdio.h>
#include<math.h>
int const N=10;
int main()
{int i,n,k;
double data[N+1],a[N+1],b[N+1],c[N+1],d[N+1];
double h,pi,s,s1;
pi=2.0*asin(1.0);      h=(2.0*pi/double(N));
//direct conversion Furie
for(i=0;i<=N-1;i++)
{
    data[i]=sin(double(i)*h);
}
for(k=0;k<=N-1;k++)
{ s=0.0;s1=0.0;
for(n=0;n<=N-1;n++)
{
    s=s+data[n]*cos(h*double(k*n));s1=s1-data[n]*sin(h*double(k*n));
}
}
a[k]=s; b[k]=s1;
}
//inverse conversion Furie
for(n=0;n<=N-1;n++)
{ s=0.0;s1=0.0;
for(k=0;k<=N-1;k++)
{
```

```

s=s+a[k]*cos(h*double(k*n))-b[k]* sin(h*double(k*n));
s1=s1+a[k]*sin(h*double(k*n))+b[k]* cos(h*double(k*n)) ;
}
c[n]=s/double(N);//d[n]=s1/double(N);
}
for(i=0;i<=N-1;i++)
{
printf("%.16f %.16f %.16f\n",data[i],c[i]);
}
for(i=0;i<=N-1;i++)
{
printf("%.16f %.16f \n",a[i],b[i]);
}}

```

Таблица 5.1. Исходный массив функции

$x_i = \sin(ih)$, $h = (b - a) / N$, $a = 0$, $b = 2\pi$, $i = 0, N - 1$, $N = 10$ записан в первом столбце, во втором и в третьем столбцах таблицы значения коэффициента $c_n, d_n, n = 0, N - 1$, то есть, после композиции прямого и обратного дискретных преобразований Фурье.

$x_i = \sin(ih)$	$c_n (5.8)$	$d_n (5.8)$
0.0000000000000000	-0.0000000000000009	0.0000000000000000
0.5877852522924731	0.5877852522924730	0.0000000000000000
0.9510565162951535	0.9510565162951536	0.0000000000000000
0.9510565162951536	0.9510565162951536	0.0000000000000000
0.5877852522924733	0.5877852522924736	0.0000000000000000
0.0000000000000001	0.0000000000000005	0.0000000000000000
-0.5877852522924730	-0.5877852522924731	0.0000000000000000
-0.9510565162951535	-0.9510565162951543	0.0000000000000000
-0.9510565162951536	-0.9510565162951536	0.0000000000000000
-0.5877852522924734	-0.5877852522924749	0.0000000000000000

Таблица 5.2. Спектр для нечетной функции $x_i = \sin(ih)$, прямое дискретное преобразование Фурье.

$A_k, k = \overline{0, N - 1}$	$B_k, k = \overline{0, N - 1}$
0.0000000000000000	0.0000000000000000
0.0000000000000001	-5.0000000000000009
0.0000000000000003	0.0000000000000001
0.0000000000000002	0.0000000000000000
0.0000000000000001	0.0000000000000001
0.0000000000000001	0.0000000000000002
0.0000000000000001	0.0000000000000003
-0.0000000000000017	0.0000000000000012
0.0000000000000002	0.0000000000000013
-0.0000000000000080	5.0000000000000018

Анализируя таблицы 5.1 и 5.2 видим, что для нечетной функции $x_i = \sin(ih)$ спектр является мнимым, действительная компонента спектра отсутствует, а после композиции

преобразований коэффициент $c_i, i = \overline{0, N-1}$ совпадает с исходным временным вектором функции x_i с двойной точностью, мнимая часть сигнала d_n равна нулю.

Пример 2. Исходная функция $x_i = \cos(ih), h = (b-a)/N, a = 0, b = 2\pi, i = \overline{0, N-1}, N = 10$ является четной

```
//furie chetnih function
#include<stdio.h>
#include<math.h>
int const N=10;
int main()
{int i,n,k;
double data[N+1],a[N+1],b[N+1],c[N+1],d[N+1];
double h,pi,s,s1;
pi=2.0*asin(1.0);      h=(2.0*pi/double(N));
//direct conversion Furie
for(i=0;i<=N-1;i++)
{
    data[i]=cos(double(i)*h);
}
for(k=0;k<=N-1;k++)
{ s=0.0;s1=0.0;
for(n=0;n<=N-1;n++)
{
    s=s+data[n]*cos(h*double(k*n));s1=s1-data[n]*sin(h*double(k*n));
}
}
a[k]=s; b[k]=s1;
}
//inverse conversion Furie
for(n=0;n<=N-1;n++)
{ s=0.0;s1=0.0;
for(k=0;k<=N-1;k++)
{
    s=s+a[k]*cos(h*double(k*n))-b[k]* sin(h*double(k*n));
    s1=s1+a[k]*sin(h*double(k*n))+b[k]* cos(h*double(k*n)) ;
}
}
c[n]=s/double(N);//d[n]=s1/double(N);
}
for(i=0;i<=N-1;i++)
{
    printf("%.16lf %.16lf %.16lf\n",data[i],c[i]);
}
printf("*****\n");
for(i=0;i<=N-1;i++)
{
    printf("%.16lf %.16lf \n",a[i],b[i]);
}
}}
```

Таблица 5.3. Исходный массив функции $x_i = \cos(ih), h = (b-a)/N, a = 0, b = 2\pi, i = \overline{0, N-1}, N = 10$ записан в первом столбце, во втором столбце таблицы значения коэффициента $c_n, n = \overline{0, N-1}$, то есть после композиции прямого и обратного дискретных преобразований Фурье.

$x_i = \cos(ih)$	$c_n(5.8)$	$d_n(5.8)$
1.0000000000000000	1.0000000000000004	0.0000000000000000
0.8090169943749475	0.8090169943749471	0.0000000000000000
0.3090169943749475	0.3090169943749470	0.0000000000000000
-0.3090169943749473	-0.3090169943749472	0.0000000000000000
-0.8090169943749473	-0.8090169943749472	0.0000000000000000
-1.0000000000000000	-1.0000000000000000	0.0000000000000000
-0.8090169943749476	-0.8090169943749471	0.0000000000000000
-0.3090169943749476	-0.3090169943749453	0.0000000000000000
0.3090169943749472	0.3090169943749473	0.0000000000000000
0.8090169943749473	0.8090169943749468	0.0000000000000000

Таблица 5.2. Спектр для четной функции $x_i = \cos(ih)$, прямое дискретное преобразование Фурье.

$A_k, k = \overline{0, N-1}$	$B_k, k = \overline{0, N-1}$
-0.0000000000000006	0.0000000000000000
5.0000000000000000	0.0000000000000000
-0.0000000000000003	0.0000000000000000
-0.0000000000000003	-0.0000000000000006
-0.0000000000000002	-0.0000000000000004
-0.0000000000000001	-0.0000000000000006
0.0000000000000002	-0.0000000000000007
0.0000000000000035	-0.0000000000000017
0.0000000000000021	-0.0000000000000010
4.9999999999999982	0.0000000000000029

Анализируя таблицы 5.3 и 5.4 видим, что для четной функции $x_i = \cos(ih)$ спектр является действительным, мнимая компонента спектра отсутствует, а после композиции преобразований коэффициент $c_i, i = \overline{0, N-1}$ совпадает с исходным временным вектором функции x_i с двойной точностью, мнимая часть сигнала d_n равна нулю.

Кроме того частотный спектр простого сигнала, мнимый или действительный $A_k, B_k, k = \overline{0, N-1}$ выражен в небольшом числе ненулевых амплитуд (для основной частоты, остальные частоты кратны основной). Это дает хороший способ для сравнения различных сигналов по их частотному спектру.

Литература

- 1) Н.С. Бахвалов, Н.П. Жидков, Г.М. Кобельков. Численные методы. – 7 – е изд.: БИНОМ. Лаборатория знаний, 2011. – 636 с. – (Классический университетский учебник).
- 2) Бахвалов Н.С., Лапин А.В., Чижонков Е.В. Численные методы в задачах и упражнениях: учебное пособие. – М.: БИНОМ. Лаборатория знаний, 2010. – 240 с.
- 3) Колмогоров А.Н., Фомин С.В. Элементы теории функции и функционального анализа. – М.: 1989 – 450 с.

Лекция 6. Двухмерное и трехмерное дискретное преобразование Фурье.

Рассмотрим прямое и обратное дискретное преобразование Фурье для двух не связанных дискретных функций $x_n^1, x_m^2, n = \overline{0, N_1 - 1}, m = \overline{0, N_2 - 1}$

$$\begin{cases} X_{k_1}^1 = \sum_{n=0}^{N_1-1} x_n^1 e^{-i\frac{2\pi k_1 n}{N_1}}, k_1 = \overline{0, N_1 - 1}, X_{k_2}^2 = \sum_{m=0}^{N_2-1} x_m^2 e^{-i\frac{2\pi k_2 m}{N_2}}, k_2 = \overline{0, N_2 - 1} \\ x_n^1 = \frac{1}{N_1} \sum_{k_1=0}^{N_1-1} X_{k_1}^1 e^{i\frac{2\pi k_1 n}{N_1}}, n = \overline{0, N_1 - 1}, x_m^2 = \frac{1}{N_2} \sum_{k_2=0}^{N_2-1} X_{k_2}^2 e^{i\frac{2\pi k_2 m}{N_2}}, m = \overline{0, N_2 - 1}. \end{cases} \quad (6.1)$$

Перемножим левые и правые части прямого и обратного преобразования Фурье в формулах (6.1), введем следующие обозначения произведения дискретных функций и их образов Фурье $x_{n,m} = x_n^1 x_m^2, n = \overline{0, N_1 - 1}, m = \overline{0, N_2 - 1}, X_{k_1, k_2} = X_{k_1}^1 X_{k_2}^2, k_1 = \overline{0, N_1 - 1}, k_2 = \overline{0, N_2 - 1}$, получим двухмерное преобразование Фурье (6.2)

$$\begin{cases} X_{k_1, k_2} = \sum_{n=0, m=0}^{N_1-1, N_2-1} x_{n,m} e^{-i2\pi \left(\frac{k_1 n}{N_1} + \frac{k_2 m}{N_2} \right)}, k_1 = \overline{0, N_1 - 1}, k_2 = \overline{0, N_2 - 1} \\ x_{n,m} = \frac{1}{N_1 N_2} \sum_{k_1=0, k_2=0}^{N_1-1, N_2-1} X_{k_1, k_2} e^{i2\pi \left(\frac{k_1 n}{N_1} + \frac{k_2 m}{N_2} \right)}, n = \overline{0, N_1 - 1}, m = \overline{0, N_2 - 1}. \end{cases} \quad (6.2)$$

Запишем также формулы прямого и обратного дискретного преобразования Фурье в симметричном виде.

$$\begin{cases} X_{k_1, k_2} = \frac{1}{\sqrt{N_1 N_2}} \sum_{n=0, m=0}^{N_1-1, N_2-1} x_{n,m} e^{-i2\pi \left(\frac{k_1 n}{N_1} + \frac{k_2 m}{N_2} \right)}, k_1 = \overline{0, N_1 - 1}, k_2 = \overline{0, N_2 - 1} \\ x_{n,m} = \frac{1}{\sqrt{N_1 N_2}} \sum_{k_1=0, k_2=0}^{N_1-1, N_2-1} X_{k_1, k_2} e^{i2\pi \left(\frac{k_1 n}{N_1} + \frac{k_2 m}{N_2} \right)}, n = \overline{0, N_1 - 1}, m = \overline{0, N_2 - 1}. \end{cases} \quad (6.3)$$

Отметим, что условие мультипликативности в общем случае могут не выполняться

$$x_{n,m} = x_n^1 x_m^2, n = \overline{0, N_1 - 1}, m = \overline{0, N_2 - 1}, X_{k_1, k_2} = X_{k_1}^1 X_{k_2}^2, k_1 = \overline{0, N_1 - 1}, k_2 = \overline{0, N_2 - 1},$$

однако формулы общего вида двухмерного дискретного преобразования Фурье по-прежнему совпадают с формулами (6.2), (6.3).

Для трехмерного преобразования Фурье требуется рассмотрение прямого и обратного дискретного Фурье для трех не связанных дискретных функций $x_n^1, x_m^2, x_s^3, n = \overline{0, N_1 - 1}, m = \overline{0, N_2 - 1}, s = \overline{0, N_3 - 1}$

$$\begin{cases} X_{k_1}^1 = \sum_{n=0}^{N_1-1} x_n^1 e^{-i\frac{2\pi k_1 n}{N_1}}, k_1 = \overline{0, N_1 - 1}, X_{k_2}^2 = \sum_{m=0}^{N_2-1} x_m^2 e^{-i\frac{2\pi k_2 m}{N_2}}, k_2 = \overline{0, N_2 - 1}, X_{k_3}^3 = \sum_{s=0}^{N_3-1} x_s^3 e^{-i\frac{2\pi k_3 s}{N_3}}, k_3 = \overline{0, N_3 - 1} \\ x_n^1 = \frac{1}{N_1} \sum_{k_1=0}^{N_1-1} X_{k_1}^1 e^{i\frac{2\pi k_1 n}{N_1}}, n = \overline{0, N_1 - 1}, x_m^2 = \frac{1}{N_2} \sum_{k_2=0}^{N_2-1} X_{k_2}^2 e^{i\frac{2\pi k_2 m}{N_2}}, m = \overline{0, N_2 - 1}, x_s^3 = \frac{1}{N_3} \sum_{k_3=0}^{N_3-1} X_{k_3}^3 e^{i\frac{2\pi k_3 s}{N_3}}, s = \overline{0, N_3 - 1}. \end{cases} \quad (6.4)$$

Перемножим левые и правые части прямого и обратного преобразования Фурье в формулах (6.4), введем следующие обозначения произведения дискретных функций и их образов

$x_{n,m,s} = x_n^1 x_m^2 x_s^3, n = \overline{0, N_1 - 1}, m = \overline{0, N_2 - 1}, s = \overline{0, N_3 - 1}, X_{k_1, k_2, k_3} = X_{k_1} X_{k_2} X_{k_3}, k_1 = \overline{0, N_1 - 1}, k_2 = \overline{0, N_2 - 1}, k_3 = \overline{0, N_3 - 1}$, получим трехмерное преобразование Фурье (6.5)

$$\begin{cases} X_{k_1, k_2, k_3} = \sum_{n=0, m=0, s=0}^{N_1-1, N_2-1, N_3-1} x_{n,m,s} e^{-i2\pi \left(\frac{k_1 n}{N_1} + \frac{k_2 m}{N_2} + \frac{k_3 s}{N_3} \right)}, k_1 = \overline{0, N_1 - 1}, k_2 = \overline{0, N_2 - 1}, k_3 = \overline{0, N_3 - 1} \\ x_{n,m,s} = \frac{1}{N_1 N_2 N_3} \sum_{k_1=0, k_2=0, k_3=0}^{N_1-1, N_2-1, N_3-1} X_{k_1, k_2, k_3} e^{i2\pi \left(\frac{k_1 n}{N_1} + \frac{k_2 m}{N_2} + \frac{k_3 s}{N_3} \right)}, n = \overline{0, N_1 - 1}, m = \overline{0, N_2 - 1}, s = \overline{0, N_3 - 1}. \end{cases} \quad (6.5)$$

Запишем также формулы прямого и обратного дискретного преобразования Фурье в симметричном виде.

$$\begin{cases} X_{k_1, k_2, k_3} = \frac{1}{\sqrt{N_1 N_2 N_3}} \sum_{n=0, m=0, s=0}^{N_1-1, N_2-1, N_3-1} x_{n,m,s} e^{-i2\pi \left(\frac{k_1 n}{N_1} + \frac{k_2 m}{N_2} + \frac{k_3 s}{N_3} \right)}, k_1 = \overline{0, N_1 - 1}, k_2 = \overline{0, N_2 - 1}, k_3 = \overline{0, N_3 - 1} \\ x_{n,m,s} = \frac{1}{\sqrt{N_1 N_2 N_3}} \sum_{k_1=0, k_2=0, k_3=0}^{N_1-1, N_2-1, N_3-1} X_{k_1, k_2, k_3} e^{i2\pi \left(\frac{k_1 n}{N_1} + \frac{k_2 m}{N_2} + \frac{k_3 s}{N_3} \right)}, n = \overline{0, N_1 - 1}, m = \overline{0, N_2 - 1}, s = \overline{0, N_3 - 1}. \end{cases} \quad (6.6)$$

Отметим, что условие мультипликативности в общем случае могут не выполняться, однако

$x_{n,m,s} = x_n^1 x_m^2 x_s^3, n = \overline{0, N_1 - 1}, m = \overline{0, N_2 - 1}, s = \overline{0, N_3 - 1}, X_{k_1, k_2, k_3} = X_{k_1} X_{k_2} X_{k_3}, k_1 = \overline{0, N_1 - 1}, k_2 = \overline{0, N_2 - 1}, k_3 = \overline{0, N_3 - 1}$

формулы общего вида трехмерного дискретного преобразования Фурье по-прежнему совпадают с формулами (6.5), (6.6).

Практическая работа 11. Двухмерное дискретное преобразование Фурье

Воспользуемся формулами (6.2) двухмерного дискретного преобразования Фурье.

$$\begin{cases} X_{k_1, k_2} = \sum_{n=0, m=0}^{N_1-1, N_2-1} x_{n,m} e^{-i2\pi \left(\frac{k_1 n}{N_1} + \frac{k_2 m}{N_2} \right)}, k_1 = \overline{0, N_1 - 1}, k_2 = \overline{0, N_2 - 1} \\ x_{n,m} = \frac{1}{N_1 N_2} \sum_{k_1=0, k_2=0}^{N_1-1, N_2-1} X_{k_1, k_2} e^{i2\pi \left(\frac{k_1 n}{N_1} + \frac{k_2 m}{N_2} \right)}, n = \overline{0, N_1 - 1}, m = \overline{0, N_2 - 1}. \end{cases}$$

Формула (6.2) может использовать как действительный временной набор данных, так и комплексный вектор $x_{n,m}, n = \overline{0, N_1 - 1}, m = \overline{0, N_2 - 1}$. Если $x_{n,m}$ действительный вектор, то из формулы (6.2) получим

$$\begin{aligned} X_{k_1, k_2} = A_{k_1, k_2} + iB_{k_1, k_2} &= \sum_{n=0, m=0}^{N_1-1, N_2-1} x_{n,m} e^{-i2\pi \left(\frac{k_1 n}{N_1} + \frac{k_2 m}{N_2} \right)} = \sum_{n=0, m=0}^{N_1-1, N_2-1} x_{n,m} \left(\cos 2\pi \left(\frac{k_1 n}{N_1} + \frac{k_2 m}{N_2} \right) - i \sin 2\pi \left(\frac{k_1 n}{N_1} + \frac{k_2 m}{N_2} \right) \right) = \\ &= A_{k_1, k_2} + iB_{k_1, k_2}, k_1 = \overline{0, N_1 - 1}, k_2 = \overline{0, N_2 - 1} \quad (6.7) \end{aligned}$$

То есть спектральные коэффициенты прямого преобразования (спектр) получим исходя из формулы (6.7)

$$X_{k_1, k_2} = \sum_{n=0, m=0}^{N_1-1, N_2-1} x_{n, m} \left(\cos 2\pi \left(\frac{k_1 n}{N_1} + \frac{k_2 m}{N_2} \right) - i \sin 2\pi \left(\frac{k_1 n}{N_1} + \frac{k_2 m}{N_2} \right) \right) \Leftrightarrow$$

$$\begin{cases} A_{k_1, k_2} = \sum_{n=0, m=0}^{N_1-1, N_2-1} x_{n, m} \cos 2\pi \left(\frac{k_1 n}{N_1} + \frac{k_2 m}{N_2} \right), k_1 = \overline{0, N_1-1}, k_2 = \overline{0, N_2-1} \\ B_{k_1, k_2} = - \sum_{n=0, m=0}^{N_1-1, N_2-1} x_{n, m} \sin 2\pi \left(\frac{k_1 n}{N_1} + \frac{k_2 m}{N_2} \right) \end{cases} \quad (6.8)$$

Используя формулы обратного дискретного преобразования Фурье, получим формулы для коэффициентов

$$x_{n, m} = c_{n, m} + i d_{n, m} = \frac{1}{N_1 N_2} \sum_{k_1=0, k_2=0}^{N_1-1, N_2-1} X_{k_1, k_2} e^{i 2\pi \left(\frac{k_1 n}{N_1} + \frac{k_2 m}{N_2} \right)} = \frac{1}{N_1 N_2} \sum_{k_1=0, k_2=0}^{N_1-1, N_2-1} (A_{k_1, k_2} + i B_{k_1, k_2}) e^{i 2\pi \left(\frac{k_1 n}{N_1} + \frac{k_2 m}{N_2} \right)} =$$

$$= \frac{1}{N_1 N_2} \sum_{k_1=0, k_2=0}^{N_1-1, N_2-1} (A_{k_1, k_2} + i B_{k_1, k_2}) \left(\cos 2\pi \left(\frac{k_1 n}{N_1} + \frac{k_2 m}{N_2} \right) - i \sin 2\pi \left(\frac{k_1 n}{N_1} + \frac{k_2 m}{N_2} \right) \right) \Leftrightarrow$$

$$\begin{cases} c_{n, m} = \frac{1}{N_1 N_2} \sum_{k_1=0, k_2=0}^{N_1-1, N_2-1} A_{k_1, k_2} \cos 2\pi \left(\frac{k_1 n}{N_1} + \frac{k_2 m}{N_2} \right) - B_{k_1, k_2} \sin 2\pi \left(\frac{k_1 n}{N_1} + \frac{k_2 m}{N_2} \right), k_1 = \overline{0, N_1-1}, k_2 = \overline{0, N_2-1} \\ d_{n, m} = \frac{1}{N_1 N_2} \sum_{k_1=0, k_2=0}^{N_1-1, N_2-1} A_{k_1, k_2} \sin 2\pi \left(\frac{k_1 n}{N_1} + \frac{k_2 m}{N_2} \right) + B_{k_1, k_2} \cos 2\pi \left(\frac{k_1 n}{N_1} + \frac{k_2 m}{N_2} \right), k_1 = \overline{0, N_1-1}, k_2 = \overline{0, N_2-1} \end{cases} \quad (6.9)$$

В итоге, коэффициенты прямого дискретного преобразования Фурье в программе запишем по формулам (6.8), а коэффициенты обратного дискретного преобразования Фурье программируем по формулам (6.9).

Опишем алгоритм работы программы на языке C++. Все функции и аргументы в программе имеют двойную точность.

Пример 1.

Исходная функция является четной относительно аргумента $w_1 n_1 + w_2 n_2$

$$x_{n_1, n_2} = \cos(w_1 n_1 + w_2 n_2), h_1 = (b-a)/n_{11}, a=0, b=2\pi, n_1 = \overline{0, n_{11}-1}, n_{11} = 5$$

$$h_2 = (d-c)/n_{22}, c=0, d=2\pi, n_2 = \overline{0, n_{22}-1}, w_1 = \frac{2\pi}{n_{11}}, w_2 = \frac{2\pi}{n_{22}}$$

Двухмерное преобразование Фурье (программа компилировалась в среде Visual Studio)

```
#include "stdafx.h"
#include <stdio.h>
#include <math.h>
int const n11=5, n22=5;
int main()
{ int i,j,k1,n1,k2,n2;
double f[n11][n22], F[n11][n22], w1, w2, pi, ff[n11][n22], a[n11][n22] ;
double b[n11][n22], s1, s2, s3, s4, c[n11][n22], d[n11][n22] ;
pi=2.0*asin(1.0);
printf("pi=% .16lf sin=% .16lf\n", 2.0*pi, sin(pi));
w1=2.0*pi/double(n11);
w2=2.0*pi/double(n22);
printf("% .16lf % .16lf\n", exp(-w1), exp(-w2));
```

```

for(n1=0;n1<=n11-1;n1++)
{
for(n2=0;n2<=n22-1;n2++)
{
    f[n1][n2]=cos(w1*n1+w2*n2);
printf(" f(%d %d)=%.16lf\n ",n1,n2, f[n1][n2]);
}}
//pryamoe
for(k1=0;k1<=n11-1;k1++)
for(k2=0;k2<=n22-1;k2++)
{
s1=0.0; s2=0.0;
for(n1=0;n1<=n11-1;n1++)
for(n2=0;n2<=n22-1;n2++)
{
s1=s1+f[n1][n2]*cos(w1*k1*n1+w2*k2*n2);s2=s2-f[n1][n2]*sin(w1*k1*n1+w2*k2*n2);
}
    a[k1][k2]=s1; b[k1][k2]=s2;
printf(" a(%d %d)=%.16lf\n ",k1,k2,a[k1][k2]); printf(" b(%d %d)=%.16lf\n ",k1,k2,b[k1][k2]);
}
//obratnoe
for(n1=0;n1<=n11-1;n1++)
for(n2=0;n2<=n22-1;n2++)
{
s1=0.0; s2=0.0;
for(k1=0;k1<=n11-1;k1++)
for(k2=0;k2<=n22-1;k2++)
{
s1=s1+a[k1][k2]*cos(w1*k1*n1+w2*k2*n2)-b[k1][k2]*sin(w1*k1*n1+w2*k2*n2);
s2=s2+ b[k1][k2]*cos(w1*k1*n1+w2*k2*n2)+a[k1][k2]*sin(w1*k1*n1+w2*k2*n2) ;
}
    ff[n1][n2]=s1/double(n11*n22);
printf(" ff(%d %d)=%.16lf mas=%.16lf\n ",n1,n2,ff[n1][n2],f[n1][n2]);
}
    return 0;
}

```

Таблица 6.1. Спектр для четной функции

$x_{n_1, n_2} = \cos(w_1 n_1 + w_2 n_2), h_1 = (b - a) / n_{11}, a = 0, b = 2\pi, n_1 = 0, n_{11} - 1, n_{11} = 5, n_2 = 0, n_{22} - 1, n_{22} = 5$
 $h_2 = (d - c) / n_{11}, c = 0, d = 2\pi, n_2 = 0, n_{22} - 1$ прямое дискретное преобразование Фурье.

k_1, k_2	A_{k_1, k_2} (6.8)	B_{k_1, k_2} (6.8)
0,0	-0.0000000000000002	0.0000000000000000
0,1	-0.0000000000000000	0.0000000000000003
0,2	-0.0000000000000001	0.0000000000000000
0,3	0.0000000000000001	0.0000000000000000
0,4	-0.0000000000000009	0.0000000000000003
1,0	-0.0000000000000005	0.0000000000000004
1,1	12.5000000000000000	-0.0000000000000003
1,2	-0.0000000000000003	0.0000000000000001
1,3	-0.0000000000000002	-0.0000000000000006
1,4	-0.0000000000000003	-0.0000000000000017
2,0	-0.0000000000000001	0.0000000000000001
2,1	0.0000000000000003	-0.0000000000000002
2,2	-0.0000000000000001	0.0000000000000001

2,3	0.0000000000000002	-0.0000000000000001
2,4	-0.0000000000000005	-0.0000000000000008
3,0	-0.0000000000000001	-0.0000000000000005
3,1	-0.0000000000000002	-0.0000000000000004
3,2	0.0000000000000000	0.0000000000000000
3,3	0.0000000000000001	-0.0000000000000001
3,4	0.0000000000000017	-0.0000000000000003
4,0	-0.0000000000000008	-0.0000000000000002
4,1	0.0000000000000001	-0.0000000000000016
4,2	0.0000000000000004	-0.0000000000000007
4,3	0.0000000000000020	-0.0000000000000012
4,4	12.5000000000000020	0.0000000000000127

Таблица 6.2. Исходный массив функций $x_{n_1, n_2} = \cos(w_1 n_1 + w_2 n_2), n_1 = \overline{0, n_{11} - 1}, n_{11} = 5, n_2 = \overline{0, n_{22} - 1}, n_{22} = 5$ записан во втором столбце, в третьем столбце таблицы значения коэффициента $c_{n_1, n_2}, n_1 = \overline{0, n_{11} - 1}, n_{11} = 5, n_2 = \overline{0, n_{22} - 1}, n_{22} = 5$, то есть после композиции прямого и обратного дискретных преобразований Фурье.

n_1, n_2	x_{n_1, n_2}	c_{n_1, n_2} (6.9)
0,0	1.0000000000000000	1.0000000000000002
0,1	0.3090169943749475	0.3090169943749477
0,2	-0.8090169943749473	-0.8090169943749472
0,3	-0.8090169943749476	-0.8090169943749479
0,4	0.3090169943749472	0.3090169943749472
1,0	0.3090169943749475	0.3090169943749477
1,1	-0.8090169943749473	-0.8090169943749476
1,2	-0.8090169943749476	-0.8090169943749476
1,3	0.3090169943749472	0.3090169943749472
1,4	1.0000000000000000	1.0000000000000000
2,0	-0.8090169943749473	-0.8090169943749475
2,1	-0.8090169943749476	-0.8090169943749476
2,2	0.3090169943749472	0.3090169943749472
2,3	1.0000000000000000	1.0000000000000000
2,4	0.3090169943749477	0.3090169943749476
3,0	-0.8090169943749476	-0.8090169943749477
3,1	0.3090169943749472	0.3090169943749472
3,2	1.0000000000000000	1.0000000000000000
3,3	0.3090169943749477	0.3090169943749477
3,4	-0.8090169943749472	-0.8090169943749476
4,0	0.3090169943749472	0.3090169943749472
4,1	1.0000000000000000	1.0000000000000000
4,2	0.3090169943749477	0.3090169943749477
4,3	-0.8090169943749472	-0.8090169943749476
4,4	-0.8090169943749477	-0.8090169943749476

Анализируя таблицы 6.1 и 6.2 видим, что для четной функции $x_{n_1, n_2} = \cos(w_1 n_1 + w_2 n_2), h_1 = (b - a) / n_{11}, n_1 = \overline{0, n_{11} - 1}, n_{11} = 5, n_2 = \overline{0, n_{22} - 1}, n_{22} = 5$ спектр

является действительным, мнимая компонента спектра отсутствует, а после композиции преобразований коэффициент c_{n_1, n_2} совпадает с исходным временным вектором функции x_{n_1, n_2} с двойной точностью, мнимая часть сигнала d_{n_1, n_2} равна нулю.

Пример 2.

Двухмерное преобразование Фурье (программа компилировалась в среде Visual Studio)

```
#include "stdafx.h"
#include <stdio.h>
#include<math.h>
int const n11=5, n22=5;
int main()
{ int i,j,k1,n1,k2,n2;
double f[n11][n22], F[n11][n22],w1, w2, pi,ff[n11][n22],a[n11][n22] ;
double b[n11][n22],s1,s2,s3,s4,c[n11][n22],d[n11][n22] ;
pi=2.0*asin(1.0);
printf("pi=% .16lf sin=% .16lf\n",2.0*pi,sin(pi));
w1=2.0*pi/double(n11);
w2=2.0*pi/double(n22);
printf("% .16lf % .16lf \n",exp(-w1),exp(-w2));
for(n1=0;n1<=n11-1;n1++)
{
for(n2=0;n2<=n22-1;n2++)
{
f[n1][n2]= sin(w1*n1)*cos(w2*n2);
printf(" f(%d %d)=% .16lf\n ",n1,n2, f[n1][n2]);
}}
//pryamoe
for(k1=0;k1<=n11-1;k1++)
for(k2=0;k2<=n22-1;k2++)
{
s1=0.0; s2=0.0;
for(n1=0;n1<=n11-1;n1++)
for(n2=0;n2<=n22-1;n2++)
{
s1=s1+f[n1][n2]*cos(w1*k1*n1+w2*k2*n2);s2=s2-f[n1][n2]*sin(w1*k1*n1+w2*k2*n2);
}
a[k1][k2]=s1; b[k1][k2]=s2;
printf(" a(%d %d)=% .16lf\n ",k1,k2,a[k1][k2]); printf(" b(%d %d)=% .16lf\n ",k1,k2,b[k1][k2]);
}
//obratnoe
for(n1=0;n1<=n11-1;n1++)
for(n2=0;n2<=n22-1;n2++)
{
s1=0.0; s2=0.0;
for(k1=0;k1<=n11-1;k1++)
for(k2=0;k2<=n22-1;k2++)
{
s1=s1+a[k1][k2]*cos(w1*k1*n1+w2*k2*n2)-b[k1][k2]*sin(w1*k1*n1+w2*k2*n2);
s2=s2+ b[k1][k2]*cos(w1*k1*n1+w2*k2*n2)+a[k1][k2]*sin(w1*k1*n1+w2*k2*n2) ;
}
ff[n1][n2]=s1/double(n11*n22);
printf(" ff(%d %d)=% .16lf mas=% .16lf\n ",n1,n2,ff[n1][n2],f[n1][n2]);
}
return 0;
}
```

Таблица 6.3. Спектр для нечетной функции $x_{n_1, n_2} = \cos(w_1 n_1) \sin(w_2 n_2), n_1 = 0, n_{11} - 1, n_{11} = 5, n_2 = 0, n_{22} - 1, n_{22} = 5$
 $h_2 = (d - c) / n_{11}, c = 0, d = 2\pi, n_2 = 0, n_{22} - 1$ прямое дискретное преобразование Фурье.

k_1, k_2	A_{k_1, k_2} (6.8)	B_{k_1, k_2} (6.8)
0,0	-0.0000000000000001	0.0000000000000000
0,1	0.0000000000000010	0.0000000000000001
0,2	0.0000000000000001	0.0000000000000002
0,3	0.0000000000000001	0.0000000000000001
0,4	0.0000000000000003	-0.0000000000000001
1,0	0.0000000000000000	0.0000000000000005
1,1	-0.0000000000000002	-6.2500000000000018
1,2	-0.0000000000000007	0.0000000000000002
1,3	-0.0000000000000010	-0.0000000000000008
1,4	0.0000000000000028	-6.2500000000000009
2,0	0.0000000000000000	0.0000000000000000
2,1	0.0000000000000005	0.0000000000000004
2,2	-0.0000000000000001	0.0000000000000001
2,3	-0.0000000000000001	0.0000000000000002
2,4	0.0000000000000003	0.0000000000000004
3,0	-0.0000000000000001	0.0000000000000002
3,1	0.0000000000000007	0.0000000000000008
3,2	-0.0000000000000001	0.0000000000000002
3,3	-0.0000000000000002	-0.0000000000000001
3,4	0.0000000000000007	0.0000000000000007
4,0	0.0000000000000001	-0.0000000000000006
4,1	-0.0000000000000034	6.2500000000000009
4,2	0.0000000000000003	-0.0000000000000004
4,3	0.0000000000000010	0.0000000000000007
4,4	-0.0000000000000061	6.2500000000000000

Таблица 6.4. Исходный массив функции $x_{n_1, n_2} = \cos(w_1 n_1) \sin(w_2 n_2), n_1 = 0, n_{11} - 1, n_{11} = 5, n_2 = 0, n_{22} - 1, n_{22} = 5$ записан во втором столбце, в третьем столбце таблицы значения коэффициента $c_{n_1, n_2}, n_1 = 0, n_{11} - 1, n_{11} = 5, n_2 = 0, n_{22} - 1, n_{22} = 5$, то есть после композиции прямого и обратного дискретных преобразований Фурье.

n_1, n_2	x_{n_1, n_2}	c_{n_1, n_2} (6.9)
0,0	0.0000000000000000	-0.0000000000000002
0,1	-0.0000000000000000	0.0000000000000000
0,2	0.0000000000000000	0.0000000000000001
0,3	0.0000000000000000	0.0000000000000001
0,4	0.0000000000000000	0.0000000000000000
1,0	0.9510565162951535	0.9510565162951536
1,1	0.2938926261462366	0.2938926261462366

1,2	-0.7694208842938133	-0.7694208842938133
1,3	-0.7694208842938134	-0.7694208842938136
1,4	0.2938926261462364	0.2938926261462366
2,0	0.5877852522924733	0.5877852522924734
2,1	0.1816356320013403	0.1816356320013404
2,2	-0.4755282581475768	-0.4755282581475770
2,3	-0.4755282581475769	-0.4755282581475770
2,4	0.1816356320013401	0.1816356320013403
3,0	-0.5877852522924730	-0.5877852522924729
3,1	-0.1816356320013402	-0.1816356320013403
3,2	0.4755282581475767	0.4755282581475767
3,3	0.4755282581475768	0.4755282581475767
3,4	-0.1816356320013401	-0.1816356320013402
4,0	-0.9510565162951536	-0.9510565162951536
4,1	-0.2938926261462366	-0.2938926261462367
4,2	0.7694208842938134	0.7694208842938134
4,3	0.7694208842938135	0.7694208842938134
4,4	-0.2938926261462364	-0.2938926261462367

Анализируя таблицы 6.3 и 6.4 видим, что для нечетной функции $x_{n_1, n_2} = \sin(w_1 n_1) \cos(w_2 n_2)$, $h_1 = (b - a) / n_{11}$, $a = 0$, $b = 2\pi$, $n_1 = 0, n_{11} - 1, n_{11} = 5$, $n_2 = 0, n_{22} - 1, n_{22} = 5$ спектр является мнимым, действительная компонента спектра отсутствует, а после композиции преобразований коэффициент c_{n_1, n_2} совпадает с исходным временным вектором функции x_{n_1, n_2} с двойной точностью, мнимая часть сигнала d_{n_1, n_2} равна нулю.

Сравнивая также спектры таблиц 6.1 и 6.3 видим, что кроме свойства четности - нечетности число ненулевых строк и их расположение в таблице спектра для разных исходных функций соответствуют различным таблицам спектра.

Практическая работа 12. Трехмерное дискретное преобразование Фурье

Воспользуемся формулами (6.6) трехмерного дискретного преобразования Фурье.

$$\left\{ \begin{array}{l} X_{k_1, k_2, k_3} = \sum_{n=0, m=0, s=0}^{N_1-1, N_2-1, N_3-1} x_{n, m, s} e^{-i2\pi \left(\frac{k_1 n}{N_1} + \frac{k_2 m}{N_2} + \frac{k_3 s}{N_3} \right)}, k_1 = \overline{0, N_1-1}, k_2 = \overline{0, N_2-1}, k_3 = \overline{0, N_3-1} \\ x_{n, m, s} = \frac{1}{N_1 N_2 N_3} \sum_{k_1=0, k_2=0, k_3=0}^{N_1-1, N_2-1, N_3-1} X_{k_1, k_2, k_3} e^{i2\pi \left(\frac{k_1 n}{N_1} + \frac{k_2 m}{N_2} + \frac{k_3 s}{N_3} \right)}, n = \overline{0, N_1-1}, m = \overline{0, N_2-1}, s = \overline{0, N_3-1} \end{array} \right.$$

Формула (6.6) может использовать как действительный временной набор данных, так и комплексный вектор $x_{n, m, s}$, $n = \overline{0, N_1-1}, m = \overline{0, N_2-1}, s = \overline{0, N_3-1}$. Если $x_{n, m, s}$ действительный вектор, то из формулы (6.6) получим, что

$$\begin{aligned} X_{k_1, k_2, k_3} &= A_{k_1, k_2, k_3} + iB_{k_1, k_2, k_3} = \sum_{n=0, m=0, s=0}^{N_1-1, N_2-1, N_3-1} x_{n, m, s} e^{-i2\pi \left(\frac{k_1 n}{N_1} + \frac{k_2 m}{N_2} + \frac{k_3 s}{N_3} \right)} = \\ &= A_{k_1, k_2, k_3} + iB_{k_1, k_2, k_3}, k_1 = \overline{0, N_1-1}, k_2 = \overline{0, N_2-1}, k_3 = \overline{0, N_3-1} \quad (6.10) \end{aligned}$$

То есть спектральные коэффициенты прямого преобразования (спектр) получим исходя из формулы (6.10)

$$X_{k_1, k_2, k_3} = \sum_{n=0, m=0, s=0}^{N_1-1, N_2-1, N_3-1} x_{n, m, s} \left(\cos 2\pi \left(\frac{k_1 n}{N_1} + \frac{k_2 m}{N_2} + \frac{k_3 s}{N_3} \right) - i \sin 2\pi \left(\frac{k_1 n}{N_1} + \frac{k_2 m}{N_2} + \frac{k_3 s}{N_3} \right) \right) \Leftrightarrow$$

$$\begin{cases} A_{k_1, k_2, k_3} = \sum_{n=0, m=0, s=0}^{N_1-1, N_2-1, N_3-1} x_{n, m, s} \cos 2\pi \left(\frac{k_1 n}{N_1} + \frac{k_2 m}{N_2} + \frac{k_3 s}{N_3} \right), k_1 = \overline{0, N_1-1}, k_2 = \overline{0, N_2-1}, k_3 = \overline{0, N_3-1} \\ B_{k_1, k_2, k_3} = - \sum_{n=0, m=0, s=0}^{N_1-1, N_2-1, N_3-1} x_{n, m, s} \sin 2\pi \left(\frac{k_1 n}{N_1} + \frac{k_2 m}{N_2} + \frac{k_3 s}{N_3} \right) \end{cases} \quad (6.11)$$

Используя формулы обратного дискретного преобразования Фурье (6.6), получим формулы для коэффициентов

$$x_{n, m, s} = c_{n, m, s} + i d_{n, m, s} = \frac{1}{N_1 N_2 N_3} \sum_{k_1=0, k_2=0, k_3=0}^{N_1-1, N_2-1, N_3-1} X_{k_1, k_2, k_3} e^{i 2\pi \left(\frac{k_1 n}{N_1} + \frac{k_2 m}{N_2} + \frac{k_3 s}{N_3} \right)} =$$

$$= \frac{1}{N_1 N_2 N_3} \sum_{k_1=0, k_2=0, k_3=0}^{N_1-1, N_2-1, N_3-1} \left(A_{k_1, k_2, k_3} + i B_{k_1, k_2, k_3} \right) e^{i 2\pi \left(\frac{k_1 n}{N_1} + \frac{k_2 m}{N_2} + \frac{k_3 s}{N_3} \right)} =$$

$$= \frac{1}{N_1 N_2 N_3} \sum_{k_1=0, k_2=0, k_3=0}^{N_1-1, N_2-1, N_3-1} \left(A_{k_1, k_2, k_3} + i B_{k_1, k_2, k_3} \right) \left(\cos 2\pi \left(\frac{k_1 n}{N_1} + \frac{k_2 m}{N_2} + \frac{k_3 s}{N_3} \right) + i \sin 2\pi \left(\frac{k_1 n}{N_1} + \frac{k_2 m}{N_2} + \frac{k_3 s}{N_3} \right) \right) \Leftrightarrow$$

$$\begin{cases} c_{n, m, s} = \frac{1}{N_1 N_2 N_3} \sum_{k_1=0, k_2=0, k_3=0}^{N_1-1, N_2-1, N_3-1} A_{k_1, k_2, k_3} \cos 2\pi \left(\frac{k_1 n}{N_1} + \frac{k_2 m}{N_2} + \frac{k_3 s}{N_3} \right) - B_{k_1, k_2, k_3} \sin 2\pi \left(\frac{k_1 n}{N_1} + \frac{k_2 m}{N_2} + \frac{k_3 s}{N_3} \right), n = \overline{0, N_1-1}, m = \overline{0, N_2-1} \\ d_{n, m, s} = \frac{1}{N_1 N_2 N_3} \sum_{k_1=0, k_2=0, k_3=0}^{N_1-1, N_2-1, N_3-1} A_{k_1, k_2, k_3} \sin 2\pi \left(\frac{k_1 n}{N_1} + \frac{k_2 m}{N_2} + \frac{k_3 s}{N_3} \right) + B_{k_1, k_2, k_3} \cos 2\pi \left(\frac{k_1 n}{N_1} + \frac{k_2 m}{N_2} + \frac{k_3 s}{N_3} \right), s = \overline{0, N_3-1} \end{cases} \quad (6.12)$$

В итоге, коэффициенты прямого дискретного преобразования Фурье в программе запишем по формулам (6.11), а коэффициенты обратного дискретного преобразования Фурье программируем по формулам (6.12).

Приведем алгоритм работы программы на языке C++. Все функции и аргументы в программе имеют двойную точность.

Пример 3.

Исходная функция является нечетной относительно аргумента $w_1 n_1$

$$x_{n_1, n_2, n_3} = \sin(w_1 n_1), h_1 = (b - a) / n_{11}, a = 0, b = 2\pi, n_1 = \overline{0, n_{11} - 1}, n_{11} = n_{22} = n_{33} = 3;$$

$$h_2 = (d - c) / n_{22}, c = e = 0, d = f = 2\pi, n_2 = \overline{0, n_{22} - 1}, h_3 = (f - e) / n_{22}; w_1 = \frac{2\pi}{n_{11}}, w_2 = \frac{2\pi}{n_{22}}, w_3 = \frac{2\pi}{n_{33}}.$$

Трехмерное преобразование Фурье (программа компилировалась в среде Visual Studio)

```
#include "stdafx.h"
#include<stdio.h>
#include<math.h>
int const n11=3, n22=3, n33=3;
int main()
{ int k1, n1, k2, n2, k3, n3;
double f[n11][n22][n33], w1, w2, pi, w3, a[n11][n22][n33], ff[n11][n22][n33];
double b[n11][n22][n33], s1, s2, s3, s4, c[n11][n22][n33], d[n11][n22][n33];
pi=2.0*asin(1.0); printf("pi=% .16lf sin=% .16lf\n", 2.0*pi, sin(pi));
```

```

w1=(2.0*pi)/double(n11);w2=(2.0*pi)/double(n22);w3=(2.0*pi)/double(n33);
printf("%.16lf %.16lf %.16lf\n",w1,w2,w3);
for(n1=0;n1<=n11-1;n1++)
for(n2=0;n2<=n22-1;n2++)
for(n3=0;n3<=n33-1;n3++)
    {
        f[n1][n2][n3]=sin(w1*double(n1));
printf(" f(%d %d %d)=%.16lf\n ",n1,n2, n3,f[n1][n2][n3]);
    }
//pryamoe
for(k1=0;k1<=n11-1;k1++)
for(k2=0;k2<=n22-1;k2++)
for(k3=0;k3<=n33-1;k3++)
    {
        s1=0.0; s2=0.0;
for(n1=0;n1<=n11-1;n1++)
for(n2=0;n2<=n22-1;n2++)
for(n3=0;n3<=n33-1;n3++)
    {
s1=s1+f[n1][n2][n3]*cos(w1*k1*n1+w2*k2*n2+w3*k3*n3);
s2=s2-f[n1][n2][n3]*sin(w1*k1*n1+w2*k2*n2+w3*k3*n3) ;
    }
        a[k1][k2][k3]=s1;    b[k1][k2][k3]=s2;
printf(" a(%d %d %d)=%.16lf b(%d %d %d)=%.16lf\n ",k1,k2,k3,a[k1][k2][k3], k1,k2,k3,b[k1][k2][k3] );
    }
printf(" %d %d %d\n",n11,n22,n33);
//obratnoe
for(n1=0;n1<=n11-1;n1++)
for(n2=0;n2<=n22-1;n2++)
for(n3=0;n3<=n33-1;n3++)
    {
        s3=0.0; s4=0.0;
for(k1=0;k1<=n11-1;k1++)
for(k2=0;k2<=n22-1;k2++)
for(k3=0;k3<=n33-1;k3++)
    {
s3=s3+(a[k1][k2][k3])*cos(w1*k1*n1+w2*k2*n2+w3*k3*n3)
(b[k1][k2][k3])*sin(w1*k1*n1+w2*k2*n2+w3*k3*n3) ;
s4=s4+(
        b[k1][k2][k3])*cos(w1*k1*n1+w2*k2*n2+w3*k3*n3)
+(a[k1][k2][k3])*sin(w1*k1*n1+w2*k2*n2+w3*k3*n3) ;
    }
        ff[n1][n2][n3]=s3/double(n11*n22*n33);    d[n1][n2][n3]=s4/double(n11*n22*n33);
printf(" c=%.16lf f=%.16lf\n ",ff[n1][n2][n3],f[n1][n2][n3]);
    }
return 0;
}

```

Таблица 6.5. Для нечетной функции применим прямое преобразование Фурье (6.11)

$$x_{n_1, n_2, n_3} = \sin(w_1 n_1), n_1 = \overline{0, n_{11} - 1}, n_2 = \overline{0, n_{22} - 1}, n_3 = \overline{0, n_{33} - 1}, n_{11} = n_{22} = n_{33} = 3;$$

k_1, k_2, k_3	A_{k_1, k_2, k_3} (6.11)	B_{k_1, k_2, k_3} (6.11)
0,0,0	0.000000000000000018	0.000000000000000000
0,0,1	0.000000000000000002	0.000000000000000003
0,0,2	0.000000000000000000	0.000000000000000000

0,1,0	0.0000000000000001	0.0000000000000004
0,1,1	0.0000000000000000	0.0000000000000002
0,1,2	0.0000000000000000	0.0000000000000000
0,2,0	0.0000000000000002	-0.0000000000000002
0,2,1	0.0000000000000000	0.0000000000000000
0,2,2	0.0000000000000001	-0.0000000000000001
1,0,0	0.0000000000000037	-13.5000000000000000
1,0,1	-0.0000000000000016	0.0000000000000010
1,0,2	-0.0000000000000042	-0.0000000000000013
1,1,0	-0.0000000000000011	0.0000000000000016
1,1,1	0.0000000000000004	0.0000000000000005
1,1,2	-0.0000000000000004	0.0000000000000000
1,2,0	-0.0000000000000044	-0.0000000000000019
1,2,1	0.0000000000000011	0.0000000000000008
1,2,2	-0.0000000000000010	-0.0000000000000004
2,0,0	-0.0000000000000100	13.5000000000000000
2,0,1	0.0000000000000027	0.0000000000000007
2,0,2	0.0000000000000032	0.0000000000000017
2,1,0	0.0000000000000031	0.0000000000000001
2,1,1	-0.0000000000000007	-0.0000000000000007
2,1,2	-0.0000000000000020	0.0000000000000017
2,2,0	0.0000000000000032	0.0000000000000011
2,2,1	-0.0000000000000007	0.0000000000000009
2,2,2	-0.0000000000000007	0.0000000000000009

Таблица 6.6. Исходный массив функции $x_{n_1, n_2, n_3} = \sin(w_1 n_1)$, $n_1 = 0, n_{11} = 1, n_2 = 0, n_{22} = 1, n_3 = 0, n_{33} = 1, n_{11} = n_{22} = n_{33} = 3$; записан во втором столбце, в третьем столбце таблицы значения коэффициента $c_{n_1, n_2, n_3}, n_1 = 0, n_{11} = 1, n_{11} = 3, n_2 = 0, n_{22} = 1, n_{22} = 3, n_3 = 0, n_{33} = 1, n_{33} = 3$, то есть после композиции прямого и обратного дискретных преобразований Фурье.

n_1, n_2, n_3	x_{n_1, n_2, n_3}	c_{n_1, n_2, n_3} (6.12)
0,0,0	0.0000000000000000	-0.0000000000000003
0,0,1	0.0000000000000000	-0.0000000000000001
0,0,2	0.0000000000000000	-0.0000000000000000
0,1,0	0.0000000000000000	-0.0000000000000002
0,1,1	0.0000000000000000	-0.0000000000000004
0,1,2	0.0000000000000000	-0.0000000000000003
0,2,0	0.0000000000000000	-0.0000000000000000
0,2,1	0.0000000000000000	-0.0000000000000002
0,2,2	0.0000000000000000	-0.0000000000000001
1,0,0	0.8660254037844387	0.8660254037844389
1,0,1	0.8660254037844387	0.8660254037844388
1,0,2	0.8660254037844387	0.8660254037844389
1,1,0	0.8660254037844387	0.8660254037844387
1,1,1	0.8660254037844387	0.8660254037844387

1,1,2	0.8660254037844387	0.8660254037844389
1,2,0	0.8660254037844387	0.8660254037844390
1,2,1	0.8660254037844387	0.8660254037844388
1,2,2	0.8660254037844387	0.8660254037844389
2,0,0	-0.8660254037844384	-0.8660254037844384
2,0,1	-0.8660254037844384	-0.8660254037844384
2,0,2	-0.8660254037844384	-0.8660254037844385
2,1,0	-0.8660254037844384	-0.8660254037844384
2,1,1	-0.8660254037844384	-0.8660254037844383
2,1,2	-0.8660254037844384	-0.8660254037844384
2,2,0	-0.8660254037844384	-0.8660254037844388
2,2,1	-0.8660254037844384	-0.8660254037844384
2,2,2	-0.8660254037844384	-0.8660254037844387

Анализируя таблицы 6.5 и 6.6 видим, что для нечетной функции $x_{n_1, n_2, n_3} = \sin(\overline{w_1 n_1}, n_1 = \overline{0, n_{11} - 1}, n_2 = \overline{0, n_{22} - 1}, n_3 = \overline{0, n_{33} - 1}, n_{11} = n_{22} = n_{33} = 3$; спектр является мнимым, действительная компонента спектра отсутствует, а после композиции преобразований коэффициент c_{n_1, n_2, n_3} совпадает с исходным временным вектором функции x_{n_1, n_2, n_3} с двойной точностью, мнимая часть сигнала d_{n_1, n_2, n_3} равна нулю.

Пример 4.

Исходная функция является четной относительно аргумента $w_3 n_3$

$$x_{n_1, n_2, n_3} = \cos(w_3 n_3), h_1 = (b - a) / n_{11}, a = 0, b = 2\pi, n_1 = \overline{0, n_{11} - 1}, n_3 = \overline{0, n_{33} - 1}, n_{11} = n_{22} = n_{33} = 3;$$

$$h_2 = (d - c) / n_{22}, c = e = 0, d = f = 2\pi, n_2 = \overline{0, n_{22} - 1}, h_3 = (f - e) / n_{22}; w_1 = \frac{2\pi}{n_{11}}, w_2 = \frac{2\pi}{n_{22}}, w_3 = \frac{2\pi}{n_{33}}.$$

Трехмерное преобразование Фурье (программа компилировалась в среде Visual Studio)

```
#include "stdafx.h"
#include<stdio.h>
#include<math.h>
int const n11=3, n22=3, n33=3;
int main()
{ int k1, n1, k2, n2, k3, n3;
double f[n11][n22][n33], w1, w2, pi, w3, a[n11][n22][n33], ff[n11][n22][n33];
double b[n11][n22][n33], s1, s2, s3, s4, c[n11][n22][n33], d[n11][n22][n33];
pi=2.0*asin(1.0); printf("pi=% .16lf sin=% .16lf\n", 2.0*pi, sin(pi));
w1=(2.0*pi)/double(n11); w2=(2.0*pi)/double(n22); w3=(2.0*pi)/double(n33);
printf("% .16lf % .16lf % .16lf\n", w1, w2, w3);
for(n1=0; n1<=n11-1; n1++)
for(n2=0; n2<=n22-1; n2++)
for(n3=0; n3<=n33-1; n3++)
{
f[n1][n2][n3]=cos(w3*double(n3));
printf(" f(%d %d %d)=% .16lf\n ", n1, n2, n3, f[n1][n2][n3]);
}
//прямоe
for(k1=0; k1<=n11-1; k1++)
for(k2=0; k2<=n22-1; k2++)
for(k3=0; k3<=n33-1; k3++)
{
s1=0.0; s2=0.0;
for(n1=0; n1<=n11-1; n1++)
```

```

for(n2=0;n2<=n22-1;n2++)
for(n3=0;n3<=n33-1;n3++)
{
s1=s1+f[n1][n2][n3]*cos(w1*k1*n1+w2*k2*n2+w3*k3*n3);
s2=s2-f[n1][n2][n3]*sin(w1*k1*n1+w2*k2*n2+w3*k3*n3) ;
}
a[k1][k2][k3]=s1;    b[k1][k2][k3]=s2;
printf(" a(%d %d %d)=%.16lf b(%d %d %d)=%.16lf \n ",k1,k2,k3,a[k1][k2][k3], k1,k2,k3,b[k1][k2][k3] );
}
printf(" %d %d %d\n",n11,n22,n33);
//obratnoe
for(n1=0;n1<=n11-1;n1++)
for(n2=0;n2<=n22-1;n2++)
for(n3=0;n3<=n33-1;n3++)
{
s3=0.0; s4=0.0;
for(k1=0;k1<=n11-1;k1++)
for(k2=0;k2<=n22-1;k2++)
for(k3=0;k3<=n33-1;k3++)
{
s3=s3+(a[k1][k2][k3])*cos(w1*k1*n1+w2*k2*n2+w3*k3*n3)
(b[k1][k2][k3])*sin(w1*k1*n1+w2*k2*n2+w3*k3*n3) ;
s4=s4+(
b[k1][k2][k3])*cos(w1*k1*n1+w2*k2*n2+w3*k3*n3)
+(a[k1][k2][k3])*sin(w1*k1*n1+w2*k2*n2+w3*k3*n3) ;
}
ff[n1][n2][n3]=s3/double(n11*n22*n33);    d[n1][n2][n3]=s4/double(n11*n22*n33);
printf(" c=%.16lf f=%.16lf\n ",ff[n1][n2][n3],f[n1][n2][n3]);
}
return 0;
}

```

Таблица 18.7. Для четной функции применим прямое преобразование Фурье (6.11)

$$x_{n_1, n_2, n_3} = \cos(w_3 n_3), n_1 = 0, n_{11} - 1, n_2 = 0, n_{22} - 1, n_3 = 0, n_{33} - 1, n_{11} = n_{22} = n_{33} = 3;$$

k_1, k_2, k_3	A_{k_1, k_2, k_3} (6.11)	B_{k_1, k_2, k_3} (6.11)
0,0,0	-0.0000000000000020	0.0000000000000000
0,0,1	13.5000000000000000	-0.0000000000000035
0,0,2	13.5000000000000000	0.0000000000000085
0,1,0	0.0000000000000000	0.0000000000000000
0,1,1	-0.0000000000000013	0.0000000000000000
0,1,2	-0.0000000000000002	0.0000000000000013
0,2,0	0.0000000000000000	-0.0000000000000001
0,2,1	0.0000000000000020	-0.0000000000000008
0,2,2	0.0000000000000013	-0.0000000000000006
1,0,0	0.0000000000000000	0.0000000000000000
1,0,1	-0.0000000000000017	-0.0000000000000002
1,0,2	-0.0000000000000014	0.0000000000000008
1,1,0	-0.0000000000000001	-0.0000000000000001
1,1,1	0.0000000000000004	0.0000000000000008
1,1,2	0.0000000000000009	0.0000000000000016
1,2,0	-0.0000000000000002	-0.0000000000000002
1,2,1	-0.0000000000000005	-0.0000000000000002
1,2,2	-0.0000000000000009	0.0000000000000019

2,0,0	0.0000000000000000	-0.0000000000000002
2,0,1	0.0000000000000022	-0.0000000000000008
2,0,2	0.0000000000000022	-0.0000000000000013
2,1,0	-0.0000000000000002	-0.0000000000000001
2,1,1	-0.0000000000000005	0.0000000000000001
2,1,2	-0.0000000000000009	0.0000000000000018
2,2,0	-0.0000000000000001	-0.0000000000000002
2,2,1	0.0000000000000001	0.0000000000000009
2,2,2	-0.0000000000000013	0.0000000000000018

Таблица 6.8. Исходный массив функций $x_{n_1, n_2, n_3} = \cos(w_3 n_3)$, $n_1 = 0, n_{11} - 1, n_2 = 0, n_{22} - 1, n_3 = 0, n_{33} - 1, n_{11} = n_{22} = n_{33} = 3$; записан во втором столбце, в третьем столбце таблицы значения коэффициента c_{n_1, n_2, n_3} , $n_1 = 0, n_{11} - 1, n_{11} = 3, n_2 = 0, n_{22} - 1, n_{22} = 3, n_3 = 0, n_{33} - 1, n_{33} = 3$, то есть после композиции прямого и обратного дискретных преобразований Фурье.

n_1, n_2, n_3	x_{n_1, n_2, n_3}	c_{n_1, n_2, n_3} (6.12)
0,0,0	1.0000000000000000	1.0000000000000004
0,0,1	-0.4999999999999998	-0.4999999999999997
0,0,2	-0.5000000000000004	-0.5000000000000008
0,1,0	1.0000000000000000	1.0000000000000002
0,1,1	-0.4999999999999998	-0.5000000000000001
0,1,2	-0.5000000000000004	-0.5000000000000003
0,2,0	1.0000000000000000	1.0000000000000002
0,2,1	-0.4999999999999998	-0.5000000000000004
0,2,2	-0.5000000000000004	-0.5000000000000001
1,0,0	1.0000000000000000	1.0000000000000002
1,0,1	-0.4999999999999998	-0.5000000000000000
1,0,2	-0.5000000000000004	-0.5000000000000004
1,1,0	1.0000000000000000	1.0000000000000000
1,1,1	-0.4999999999999998	-0.4999999999999997
1,1,2	-0.5000000000000004	-0.5000000000000003
1,2,0	1.0000000000000000	1.0000000000000000
1,2,1	-0.4999999999999998	-0.4999999999999999
1,2,2	-0.5000000000000004	-0.5000000000000004
2,0,0	1.0000000000000000	1.0000000000000002
2,0,1	-0.4999999999999998	-0.5000000000000001
2,0,2	-0.5000000000000004	-0.5000000000000004
2,1,0	1.0000000000000000	1.0000000000000000
2,1,1	-0.4999999999999998	-0.4999999999999998
2,1,2	-0.5000000000000004	-0.5000000000000006
2,2,0	1.0000000000000000	1.0000000000000000
2,2,1	-0.4999999999999998	-0.4999999999999999
2,2,2	-0.5000000000000004	-0.5000000000000004

Анализ таблиц 6.7 и 6.8 показывает, что для четной функции $x_{n_1, n_2, n_3} = \cos(w_3 n_1)$, $n_1 = 0, n_{11} - 1, n_2 = 0, n_{22} - 1, n_3 = 0, n_{33} - 1, n_{11} = n_{22} = n_{33} = 3$; спектр является действительным, а после композиции преобразований коэффициент c_{n_1, n_2, n_3} совпадает с

исходным временным вектором функции x_{n_1, n_2, n_3} с двойной точностью, мнимая часть сигнала d_{n_1, n_2, n_3} равна нулю.

Анализ четырех примеров для двухмерного и трехмерного дискретного преобразований Фурье и результатов таблиц 6.1, 6.2, 6.3, 6.4, 6.5, 6.6, 6.7, 6.8 позволяет сделать **основные выводы**:

1) Для функции нескольких переменных можно обобщить понятие четной или нечетной функции

Определение 1. Говорят, что функция нескольких переменных $f(x_1, x_2, \dots, x_n)$ является четной, если

$$f(-x_1, -x_2, \dots, -x_n) = f(x_1, x_2, \dots, x_n) \quad (6.13)$$

Определение 2. Говорят, что функция нескольких переменных $f(x_1, x_2, \dots, x_n)$ является нечетной, если

$$f(-x_1, -x_2, \dots, -x_n) = -f(x_1, x_2, \dots, x_n) \quad (6.14)$$

2) Понятие четной или нечетной функции для дискретной функции, заданной по каждой переменной с равномерным шагом, можно обобщить тогда и только тогда, когда узлы и узловые значения функции заданы на периоде данной функции по каждой из переменных (например, с периодом 2π по каждой переменной). Например, если дискретный входной периодический сигнал нескольких натуральных индексов для дискретного Фурье - преобразования можно задать формулой $n_i = \overline{0, N_i - 1}, i = \overline{1, m}$.

$$x_{n_1, n_2, \dots, n_m} = x(\omega_1 n_1 + \omega_2 n_2 + \dots + \omega_m n_m), \omega_i = \frac{2\pi}{N_i}, n_i = \overline{0, N_i - 1}, i = \overline{1, m} \quad (6.15)$$

Литература

- 1) Н.С. Бахвалов, Н.П. Жидков, Г.М. Кобельков. Численные методы. – 7 – е изд.: БИНОМ. Лаборатория знаний, 2011. – 636 с. – (Классический университетский учебник).
- 2) Бахвалов Н.С., Лапин А.В., Чижонков Е.В. Численные методы в задачах и упражнениях: учебное пособие. – М.: БИНОМ. Лаборатория знаний, 2010. – 240 с.
- 3) Колмогоров А.Н., Фомин С.В. Элементы теории функции и функционального анализа. – М.: 1989 – 450 с.

ЛИТЕРАТУРА

1. Тихомиров, В. М. Радость математического открытия к 120-летию со дня рождения академика А.Н. Колмогорова / В. М. Тихомиров // Вестник Российской академии наук. – 2023. – Т. 93, № 4. – С. 373-383. – DOI 10.31857/S0869587323040126. – EDN SDTHVT.
2. Тихомиров, В. М. Сергей Борисович Стечкин (1920-1995) / В. М. Тихомиров, Г. Г. Магарил-Ильяев // Вестник Владикавказского научного центра. – 2021. – Т. 21, № 4. – С. 85. – EDN VOJJEQ.
3. Алексеев, В. М. Сборник задач по оптимизации: Теория. Примеры. Задачи / В. М. Алексеев, Э. М. Галеев, В. М. Тихомиров. – Москва: Физматлит, 2008. – 256 с. – ISBN 978-5-9221-0992-5. – EDN QEAJNP.
4. Бахвалов, Н. С. Численные методы: учебное пособие для студентов физ.-мат. специальностей вузов / Н. С. Бахвалов, Н. П. Жидков, Г. М. Кобельков; Н. С. Бахвалов, Н. П. Жидков, Г. М. Кобельков. – 3-е изд. доп. и перераб.. Москва: Бином. Лаб. знаний, 2003. – 632 с. – (Математика/Техн. ун-т). – ISBN 5-94774-060-5.
5. Бахвалов, Н. С. Численные методы в задачах и упражнениях. Учебное пособие / Н. С. Бахвалов, А. В. Лапин, Е. В. Чижонков. – Москва: БИНОМ. Лаборатория знаний, 2010. – 241 с. – ISBN 978-5-9963-0333-5. – EDN RBARWH.
6. Бахвалов, Н. С. Численные методы: решения задач и упражнения : учебное пособие для студентов высших учебных заведений, обучающихся по специальности "010100 Математика" / Н. С. Бахвалов, А. А. Корнев, Е. В. Чижонков ; Н. С. Бахвалов, А. А. Корнев, Е. В. Чижонков. – Москва : Дрофа, 2009. – (Высшее образование : современный учебник). – ISBN 978-5-358-03610-9. – EDN QJUTOB.
7. Кобельков, Г. М. Актуальные задачи вычислительной математики / Г. М. Кобельков // Вестник Московского университета. Серия 1: Математика. Механика. – 2024. – № 6. – С. 8-15. – DOI 10.55959/MSU0579-9368-1-65-6-2. – EDN FNYHXQ.
8. Kobelkov, G. M. Topical Problems of Computational Mathematics / G. M. Kobelkov // Moscow University Mathematics Bulletin. – 2024. – Vol. 79, No. 6. – P. 315-321. – DOI 10.3103/S0027132225700020. – EDN AXUBRS.
9. Kobelkov, G. M. Connection between the existence of a priori estimate for a flux and the convergence of iterative methods for diffusion equation with highly varying coefficients / G. M. Kobelkov, E. Schnack // Russian Journal of Numerical Analysis and Mathematical Modelling. – 2022. – Vol. 37, No. 3. – P. 143-147. – DOI 10.1515/rnam-2022-0012. – EDN ZTXHGZ.
10. Применение нейронных сетей для распознавания конформационных изменений в структуре белка по рентгеновским дифрактограммам его одиночных молекул на примере фотоцикла бактериородопсина / Г. А. Армеев, М. П. Кирпичников, Г. М. Кобельков [и др.] // Интеллектуальные системы. Теория и приложения. – 2022. – Т. 26, № 1. – С. 24-34. – EDN WIPSCX.
11. Конев, С. В. Применение нейросетевых технологий для диагностики неисправностей авиационных двигателей / С. В. Конев, З. И. Сичинава, Л. Н. Ясницкий // Вестник Пермского университета. Математика. Механика. Информатика. – 2005. – № 2(2). – С. 43-47. – EDN YMWFWP.

12. Петров, А. М. Возможности создания нейросетевого полиграфа / А. М. Петров, Л. Н. Ясницкий // Вестник Пермского университета. Математика. Механика. Информатика. – 2005. – № 2(2). – С. 97-98. – EDN UIWPVO.
13. Черепанов, Ф. М. Нейросетевой фильтр для исключения выбросов в статистической информации / Ф. М. Черепанов, Л. Н. Ясницкий // Вестник Пермского университета. Математика. Механика. Информатика. – 2008. – № 4(20). – С. 151-155. – EDN MNLKFX.
14. Стаценко, Н. С. Онтологическое моделирование в практике современного программирования на примере автоматизации исследований в области искусственных нейронных сетей / Н. С. Стаценко, К. А. Юрков // Вестник Пермского университета. Математика. Механика. Информатика. – 2009. – № 3(29). – С. 157-160. – EDN KHNUXL.
15. Кирилловых, С. А. Нейросетевой подход к реализации агента анализа подсистемы балансировки TriadBalance / С. А. Кирилловых, К. А. Юрков // Вестник Пермского университета. Математика. Механика. Информатика. – 2010. – № 1(1). – С. 98-105. – EDN LGKJTJ.
16. Чернопятов, А. В. Применение методов искусственного интеллекта в системах видеонаблюдения / А. В. Чернопятов // Вестник Пермского университета. Математика. Механика. Информатика. – 2010. – № 4(4). – С. 54-58. – EDN NDXQWH.
17. Опыт создания нейросетевой системы для диагностики сердечно-сосудистых заболеваний / А. А. Думлер, А. Н. Полещук, К. В. Богданов [и др.] // Вестник Пермского университета. Математика. Механика. Информатика. – 2011. – № 1(5). – С. 95-101. – EDN NTUHYV.
18. Баталов, А. С. Методы повышения эффективности обучения нейронной сети Кохонена / А. С. Баталов // Вестник Пермского университета. Математика. Механика. Информатика. – 2012. – № 3(11). – С. 86-93. – EDN PJHXCN.
19. Черепанов, Ф. М. Симулятор нейронных сетей для вузов / Ф. М. Черепанов // Вестник Пермского университета. Математика. Механика. Информатика. – 2012. – № 3(11). – С. 98-105. – EDN PJHXDH.
20. Чернопятов, А. В. Нейросетевая система распознавания номеров железнодорожных вагонов / А. В. Чернопятов // Вестник Пермского университета. Математика. Механика. Информатика. – 2013. – № 4(23). – С. 96-102. – EDN RPYHBP.
21. Методика нейросетевого прогнозирования результатов спортивных состязаний на примере чемпионата мира-2015 по легкой атлетике / Л. Н. Ясницкий, А. В. Кирсова, А. В. Ратегова, Ф. М. Черепанов // Вестник Пермского университета. Математика. Механика. Информатика. – 2014. – № 3(26). – С. 90-97. – EDN SXTVMB.
22. Ханжина, Н. Е. Использование классических методов и нейронных сетей для распознавания пыльцевых зерен / Н. Е. Ханжина, Е. Б. Замятина // Вестник Пермского университета. Математика. Механика. Информатика. – 2014. – № 4(27). – С. 111-119. – EDN TCRTWF.
23. Применение нейросетевых технологий в изучении акмеологического потенциала студентов вуза / Л. Н. Ясницкий, А. Г. Кузнецов, С. М. Селезнева [и др.] // Вестник Пермского университета. Математика. Механика. Информатика. – 2014. – № 4(27). – С. 120-126. – EDN TCRTWP.
24. Лавриков, Д. С. Использование методов искусственного интеллекта для оценки состояния мышц ног при выполнении приседаний / Д. С. Лавриков, Е. Г. Корякин //

- Вестник Пермского университета. Математика. Механика. Информатика. – 2016. – № 4(35). – С. 53-57. – DOI 10.17072/1993-0550-2016-4-53-57. – EDN XUXJKJ.
25. Старкова, В. О. Опыт применения методов искусственного интеллекта для выявления пользователей социальной сети, готовых к совершению противоправных действий / В. О. Старкова, Е. Ю. Никитина // Вестник Пермского университета. Математика. Механика. Информатика. – 2020. – № 1(48). – С. 80-86. – DOI 10.17072/1993-0550-2020-1-80-86. – EDN QLJJCW.
26. Бутина, Д. В. Нейросетевая система прогнозирования результатов сезона итальянской футбольной лиги "Серия А" / Д. В. Бутина, Л. Н. Ясницкий // Вестник Пермского университета. Математика. Механика. Информатика. – 2023. – № 1(60). – С. 84-92. – DOI 10.17072/1993-0550-2023-1-84-92. – EDN EWRPRO.
27. Иванова, К. В. Искусственный интеллект для контроля передачи данных в тактическом звене управления с использованием многослойного и многопоточного шифрования геопространственной обстановки / К. В. Иванова, А. Ф. Сальников, Р. В. Мормуль // Вестник Пермского университета. Математика. Механика. Информатика. – 2023. – № 2(61). – С. 65-71. – DOI 10.17072/1993-0550-2023-2-65-71. – EDN OMLBWW.
28. Быков, И. В. Применение математических методов в искусственной нейронной сети для распознавания в голосе "фальшивых" эмоций / И. В. Быков, Н. Ю. Ротанева, А. П. Шкарапуга // Вестник Пермского университета. Математика. Механика. Информатика. – 2024. – № 4(67). – С. 104-116. – DOI 10.17072/1993-0550-2024-4-104-116. – EDN ENFGRE.
29. Применение искусственного интеллекта для привлечения клиентов в социальных сетях: обзор российского и зарубежного опыта / А. В. Соколов, А. С. Шуткин, Е. М. Епифанова [и др.] // Вестник Пермского университета. Математика. Механика. Информатика. – 2025. – № 1(68). – С. 118-144. – DOI 10.17072/1993-0550-2025-1-118-144. – EDN PQGFHJ.
30. Модели сегментации сердечных артерий по коронарографическим снимкам / В. А. Бочкарев, А. А. Усынин, А. Д. Осипов [и др.] // Вестник Пермского университета. Математика. Механика. Информатика. – 2025. – № 2(69). – С. 65-87. – DOI 10.17072/1993-0550-2025-2-65-87. – EDN SIJPKL.
31. Захарова, А. А. Применение метода вычитания фона и алгоритма Виолы-Джонса для увеличения скорости трекинга лиц в системе распознавания личности / А. А. Захарова, С. Г. Небаба // Computational Mathematics and Information Technologies. – 2018. – Т. 2, № 1. – С. 25-32. – DOI 10.23947/2587-8999-2018-2-1-25-32. – EDN YVEKOD.
32. Машинное обучение в анализе влияния электромагнитного поля на скорость коррозии и солеотложения нефтепромыслового оборудования / Ш. Р. Хуснуллин, К. Ф. Коледина, С. Р. Алимбекова, Ф. Г. Ишмуратов // Computational Mathematics and Information Technologies. – 2023. – Т. 6, № 1. – С. 70-76. – DOI 10.23947/2587-8999-2023-6-1-70-76. – EDN GSODCI.
33. Галабурдин, А. В. Применение нейронных сетей для решения задачи Дирихле для областей сложной формы / А. В. Галабурдин // Computational Mathematics and Information Technologies. – 2024. – Т. 8, № 2. – С. 68-79. – DOI 10.23947/2587-8999-2024-8-2-68-79. – EDN NNPUQB.

34. Галабурдин, А. В. Применение нейронных сетей для решения нелинейных краевых задач для областей сложной формы / А. В. Галабурдин // *Computational Mathematics and Information Technologies*. – 2024. – Т. 8, № 4. – С. 35-42. – DOI 10.23947/2587-8999-2024-8-4-35-42. – EDN VMMGIT.
35. Сидорякина, В. В. Идентификация морских разливов нефти на основе нейросетевых технологий / В. В. Сидорякина, Д. А. Соломаха // *Computational Mathematics and Information Technologies*. – 2024. – Т. 8, № 4. – С. 43-48. – DOI 10.23947/2587-8999-2024-8-4-43-48. – EDN OCZNUH.
36. Рахимбаева, Е. О. Автоматическое распознавание значений глубины на лоцманских картах с использованием методов глубокого обучения / Е. О. Рахимбаева, Т. А. Алышов, Ю. В. Белова // *Computational Mathematics and Information Technologies*. – 2025. – Т. 9, № 1. – С. 52-60. – DOI 10.23947/2587-8999-2025-9-1-52-60. – EDN YLMJWA.
37. Галабурдин, А. В. Применение нейронных сетей при решении эллиптических уравнений для областей сложной формы / А. В. Галабурдин // *Computational Mathematics and Information Technologies*. – 2025. – Т. 9, № 2. – С. 44-51. – DOI 10.23947/2587-8999-2025-9-2-44-51. – EDN BBQMSK.
38. Автоматическая обработка первичных данных натурных исследований поведения природно-технических систем при изменении климата и антропогенных воздействиях в условиях Крайнего Севера / С. В. Кушуков, К. Н. Иванов, С. П. Левашкин, М. В. Якововский // *Computational Mathematics and Information Technologies*. – 2025. – Т. 9, № 2. – С. 52-64. – DOI 10.23947/2587-8999-2025-9-2-52-64. – EDN DMEKOC.
39. Галабурдин, А. В. Применение нейронных сетей для решения задачи об установившихся колебаниях / А. В. Галабурдин // *Computational Mathematics and Information Technologies*. – 2025. – Т. 9, № 3. – С. 56-63. – DOI 10.23947/2587-8999-2025-9-3-56-63. – EDN PVWQCR.
40. Гибридное моделирование экстремальных штормовых процессов и рисков судоходства в Азовском море на основе трёхмерной гидродинамики и методов машинного обучения / А. И. Сухинов, С. В. Проценко, Е. А. Проценко, Н. Д. Панасенко // *Computational Mathematics and Information Technologies*. – 2025. – Т. 9, № 4. – С. 10-21. – DOI 10.23947/2587-8999-2025-9-4-10-21. – EDN WGYMYU.
41. Галабурдин, А. В. Аппроксимация функций нейронными сетями / А. В. Галабурдин // *Computational Mathematics and Information Technologies*. – 2026. – Т. 10, № 1. – С. 50-57. – DOI 10.23947/2587-8999-2026-10-1-50-57. – EDN СТВЗВЕ.
42. Сухинов, А. И. Обнаружение разливов нефти на основе усовершенствованных LBP-нейроалгоритмов для зашумленных космических снимков / А. И. Сухинов, Д. А. Соломаха, В. В. Сидорякина // *Computational Mathematics and Information Technologies*. – 2026. – Т. 10, № 1. – С. 58-71. – DOI 10.23947/2587-8999-2026-10-1-58-71. – EDN DKNXKX.
43. Нгуен, Т. Д. Нейросетевой метод решения краевой задачи для дифференциальных уравнений дробного порядка / Т. Д. Нгуен // *Вычислительные методы и программирование*. – 2025. – Т. 26, № 3. – С. 245-253. – DOI 10.26089/NumMet.v26r317. – EDN XDКМОО.
44. Сухинов, А. И. Усовершенствованный метод распознавания объектов морских и прибрежных систем на основе комбинации метода локальных бинарных шаблонов и нейросетевых технологий / А. И. Сухинов, Д. А. Соломаха // *Вычислительные методы*

- и программирование. – 2025. – Т. 26, № 3. – С. 366-379. – DOI 10.26089/NumMet.v26r324. – EDN TPDPEZ.
45. Методика анализа производительности вывода глубоких нейронных сетей на примере задачи классификации изображений / М. Р. Алибеков, Н. Е. Березина, Е. П. Васильев [и др.] // Вычислительные методы и программирование. – 2024. – Т. 25, № 2. – С. 127-141. – DOI 10.26089/NumMet.v25r211. – EDN LGJZFS.
46. Оптимизация тренировочного набора данных для подавляющей численную дисперсию нейронной сети NDM-net / Е. А. Гондюл, В. В. Лисица, К. Г. Гадыльшин, Д. М. Вишневский // Вычислительные методы и программирование. – 2024. – Т. 25, № 2. – С. 155-174. – DOI 10.26089/NumMet.v25r213. – EDN СЕЕУОУ.
47. Голубев, В. И. Опыт применения сверточных нейронных сетей к обратным задачам сейсмической разведки / В. И. Голубев, М. И. Анисимов // Вычислительные методы и программирование. – 2024. – Т. 25, № 5. – С. 46-61. – DOI 10.26089/NumMet.2024s04. – EDN TМFPSI.
48. Цымблер, М. Л. Восстановление пропущенных значений временного ряда на основе совместного применения аналитических алгоритмов и нейронных сетей / М. Л. Цымблер, А. А. Юртин // Вычислительные методы и программирование. – 2023. – Т. 24, № 3. – С. 243-259. – DOI 10.26089/NumMet.v24r318. – EDN ВJZYCZ.
49. Множества с внешним чебышевским слоем. Алимов А.Р., Карлов М.И. Математические заметки. 2001. Т. 69. № 2. С. 303.
50. Alimov, A. R. Sets with external Chebyshev layer / A. R. Alimov, M. I. Karlov // Mathematical Notes. – 2001. – Vol. 69, No. 1-2. – P. 269-273. – DOI 10.1023/a:1002836705675. – EDN ROPRFV.
51. ЭКСТРЕМАЛЬНЫЕ ЗАДАЧИ ТЕОРИИ ФУНКЦИЙ, ТЕОРИИ ПРИБЛИЖЕНИЙ И ГЕОМЕТРИИ КОНЯГИН С.В., ЦАРЬКОВ И.Г., ИВАНОВСКИЙ С.Б., КАРАПЕТЯН А.Г., КАРЛОВ М.И., КОЗКО А.И., МОЩЕВИТИН Н.Г., ПОПОВ А.Ю., ПРОТАСОВ В.Ю., РАЙГОРОДСКИЙ А.М., РЮТИН К.С., ТИМОШИН О.А. – EDN АЕУМОZ.
52. МНОЖЕСТВА С ВНЕШНИМ ЧЕБЫШЕВСКИМ СЛОЕМ Алимов А.Р., Карлов М.И. Математические заметки. 2001. Т. 69. № 2. С. 303.
53. CONVEXITY AND CONNECTEDNESS OF CHEBYSHEV SETS AND SUNS Karlov M.I., Tsar'kov I.G. Фундаментальная и прикладная Математика. 1997. Т. 3. № 4. С. 967.
54. CHEBYSHEV LAYER OF MANIFOLDS IN A HILBERT SPACE Karlov M.I. Proceedings of the Steklov Institute of Mathematics. 1997. Т. 219. С. 231.
55. Шапаева, Т. Б. Исследование динамики доменных границ в прозрачных ферромагнетиках методом высокоскоростной фотографии / Т. Б. Шапаева // Вестник Московского университета. Серия 3: Физика. Астрономия. – 2024. – № 6. – С. 2460506. – DOI 10.55959/MSU0579-9392.79.2460506. – EDN SMYFTF.
56. Измерение подвижности доменных границ в аморфных лентах NiCoFeSiB с помощью Керр-микроскопии / С. В. Самченко, Н. Н. Перова, Т. Б. Шапаева, Н. С. Перов // Ученые записки физического факультета Московского университета. – 2024. – № 4. – С. 2441701. – EDN AUCXOI.
57. Shapaeva, T. B. “Gray” slit diffraction and the possibility of its use to improve the spatial resolution of magneto-optics / T. B. Shapaeva, E. F. Khudaigulova // Journal of Magnetism and Magnetic Materials. – 2023. – Vol. 587. – P. 171369. – DOI 10.1016/j.jmmm.2023.171369. – EDN EBYIFJ.

58. Исследование динамики доменной границы в GdFeCo методом двукратной высокоскоростной фотографии / К. Н. Prabhakara, Т. Б. Шапаева, В. В. Юрлов [и др.] // Физика твердого тела. – 2023. – Т. 65, № 2. – С. 248-253. – DOI 10.21883/FTT.2023.02.54298.506. – EDN JGCLOW.
59. Magneto-optical diffraction of visible light as a probe of nanoscale displacement of domain walls at femtosecond timescales / A. Dolgikh, Т. В. Shapaeva, К. Т. Yamada [et al.] // Review of Scientific Instruments. – 2023. – Vol. 94, No. 10. – DOI 10.1063/5.0152670. – EDN RPHMYT.
60. Влияние параметров управляющего импульса магнитного поля на динамику доменной границы / Т. Б. Шапаева, А. Р. Юмагузин, Ю. Н. Курбатова, Р. М. Вахитов // Физика металлов и металловедение. – 2022. – Т. 123, № 3. – С. 284-290. – DOI 10.31857/S0015323022030111. – EDN IMXNSA.
61. Шапаева, Т. Б. Аппроксимация экспериментальных данных по динамике антиферромагнитных вихрей в доменной границе ортоферрита иттрия / Т. Б. Шапаева, Э. Ф. Худайгулова // Ученые записки физического факультета Московского университета. – 2022. – № 6. – С. 2260501. – EDN AUFSEZ.
62. Study of Domain Wall Dynamics in GdFeCo Using Double High-Speed Photography / К. Н. Prabhakara, Т. В. Shapaeva, V. V. Yurlov [et al.] // Physics of the Solid State. – 2022. – Vol. 64, No. 10. – P. 547-553. – DOI 10.1134/s1063783422110105. – EDN NMWKRI.
63. Торможение доменной границы в пленке феррита-граната с помощью фемтосекундных лазерных импульсов / К. Х. Прабхакара, Т. Б. Шапаева, М. Д. Давыдова [и др.] // Вестник Московского университета. Серия 3: Физика. Астрономия. – 2021. – № 6. – С. 33-40. – EDN ВАUQHZ.
64. Шапаева, Т. Б. Подвижность доменной границы феррита-граната в присутствии плоскостного магнитного поля / Т. Б. Шапаева, Ю. Н. Курбатова // Краткие сообщения по физике ФИАН. – 2021. – Т. 48, № 8. – С. 17-23. – EDN FRNPVO.
65. Шапаева, Т. Б. Подвижность доменной границы феррита-граната в присутствии плоскостного магнитного поля / Т. Б. Шапаева, Ю. Н. Курбатова // Краткие сообщения по физике ФИАН. – 2021. – Т. 48, № 8. – С. 17-23. – EDN FRNPVO.
66. Optical Excitation and Probing of Antiferromagnetic Modes with Nonuniform-in-depth Distribution in Birefringent Antiferromagnetic Crystals / А. А. Voronov, D. O. Ignatyeva, А. К. Zvezdin [et al.] // Physical Review Applied. – 2021. – Vol. 16, No. 4. – P. L051001. – DOI 10.1103/PhysRevApplied.16.L051001. – EDN UYGNBE.
67. Chetkin, M. V. Peculiarity of solitary deflection waves dynamics on the domain walls of yttrium orthoferrite / M. V. Chetkin, Y. N. Kurbatova, Т. В. Shapaeva // Solid State Phenomena. – 2015. – Vol. 233-234. – P. 435-438. – DOI 10.4028/www.scientific.net/SSP.233-234.435. – EDN VANMTP.
68. Волосов, К. А. О собственных функциях структур, описываемых моделью “мелкой воды” на плоскости / К.А. Волосов// Фундаментальная и прикладная математика. 2006. Т. 12. № 6. С. 17-32. EDN ILMIGY.
69. Волосов, К. А. Одевание решений для некоторых неинтегрируемых задач и некоторые инвариантные свойства анзаца метода Хироты / К. А. Волосов// Дифференциальные уравнения. 2005. – Т. 41, № 11. – С. 1572-1575. – EDN HSGHND.
70. Волосов, К. А. Построение решений квазилинейных параболических уравнений в параметрической форме / К. А. Волосов// Дифференциальные уравнения. – 2007 . Т.43, № 4. – С. 492-497. – EDN IAGXXJ.

71. Волосов, К. А. Новый метод построения решений уравнений с частными производными в параметрической форме / К. А. Волосов // Известия Российского государственного педагогического университета им. А.И. Герцена. – 2007. Т. 7, № 26. С. 13-20. – EDN NRTQCH.
72. Волосов, К. А. Конструкция решений квазилинейных уравнений с частными производными / К. А. Волосов // Сибирский журнал индустриальной математики 2008. – Т. 11, № 2(34). – С. 29-39. – EDN IUDSGT.
73. Волосова, А. К. Интегрирование уравнений Гарри Дима и Кортевега де Вриза в параметрической форме / А. К. Волосова. К. А. Волосов // Дифференциальные уравнения и процессы управления. – 2017. – № 4. – С. 194-214. – EDN ZXNEVZ.
74. Вдовина Е. К. Моделирование пульсирующих режимов динамики свертывания крови / Е. К. Вдовина, Л. В. Пугина, К. А. Волосов // Математическое моделирование. – 2014. Т. 26, № 12. С. – 14-32. – EDN TSNVEJ.
75. Вдовина Е. К. Моделирование спиральных волн в процессе свертывания крови / Е. К. Вдовина, К. А. Волосов // Математическое моделирование. 2013. – Т. 25, № 3. – С. 14-24. – EDN RXPQHV.
76. Волосов, К. А. Новый метод построения решений уравнений с частными производными в параметрической форме / К. А. Волосов // Известия Российского государственного педагогического университета им. А.И. Герцена. – 2007. – Т. 7, № 26. – С. 13-20. – EDN NRTQCH.
77. Волосов, К. А. Методика анализа эволюционных систем с распределенными параметрами : специальность 05.13.01 "Системный анализ, управление и обработка информации (по отраслям)" : диссертация на соискание ученой степени доктора физико-математических наук / Волосов Константин Александрович. – Москва, 2007. – 277 с. – EDN QDYHQN.
78. Вдовина, Е. К. Моделирование пульсирующих режимов динамики свертывания крови / Е. К. Вдовина, Л. В. Пугина, К. А. Волосов // Математическое моделирование. – 2014. – Т. 26, № 12. – С. 14-32. – EDN TSNVEJ.
79. Критерий существования решений, количество решений, структура решений канонического неоднородного алгебраического уравнения степени n в кольце вычетов Z_m с генераторами и свободным членом, взаимно простым с порядком кольца / Ю. Ф. Пастухов, Д. Ф. Пастухов, С. В. Чернов [и др.] // Тенденции развития науки и образования. – 2023. – № 101-4. – С. 114-117. – DOI 10.18411/trnio-09-2023-186. – EDN ELNFXM.
80. Козлов, А. А. О свойстве равномерной полной управляемости для линейной системы с локально интегрируемыми коэффициентами / А. А. Козлов // Вестник Полоцкого государственного университета. Серия С. Фундаментальные науки. – 2024. – № 2(43). – С. 62-66. – DOI 10.52928/2070-1624-2024-43-2-62-66. – EDN KXIMIQ.
81. Козлов, А. А. Равномерная глобальная достижимость дискретных периодических систем / А. А. Козлов // Дифференциальные уравнения. – 2023. – Т. 59, № 2. – С. 289-290. – EDN CVTRQS.
82. Козлов, А. А. Факторизация квадратных матриц с ненулевыми главными угловыми минорами и положительным определителем строго положительно регулярными матрицами / А. А. Козлов, Т. А. Александрович // Веснік Віцебскага дзяржаўнага ўніверсітэта. – 2023. – № 1(118). – С. 5-20. – EDN OHRJSM.
83. Козлов, А. А. О свойствах строго положительно регулярных матриц / А. А. Козлов, Т. А. Александрович // Веснік Віцебскага дзяржаўнага ўніверсітэта. – 2022. – № 4(117). – С. 5-16. – EDN TDZBHX.

84. Козлов, А. А. Множество полуоктав. II / А. А. Козлов // Вестник Полоцкого государственного университета. Серия С. Фундаментальные науки. – 2022. – № 4. – С. 103-111. – DOI 10.52928/2070-1624-2022-38-4-103-111. – EDN IPIVJB.
85. Козлов, А. А. Критерий равномерной глобальной достижимости периодических систем / А. А. Козлов // Вестник Удмуртского университета. Математика. Механика. Компьютерные науки. – 2020. – Т. 30, № 2. – С. 221-236. – DOI 10.35634/vm200206. – EDN TMYZZB.
86. Козлов, А. А. Равномерная полная управляемость линейных дискретных систем с изменяющейся структурой / А. А. Козлов, Т. А. Александрович // Веснік Віцебскага дзяржаўнага ўніверсітэта. – 2020. – № 4(109). – С. 20-27. – EDN EVXOCK.
87. Козлов, А. А. Преобразование подобия во множестве полукватернионов / А. А. Козлов, К. С. Суравнева, Н. Д. Жалейко // Вестник Полоцкого государственного университета. Серия С. Фундаментальные науки. – 2019. – № 4. – С. 115-123. – EDN SDWINI.
88. Kozlov, A. A. Control over some asymptotic invariants of two-dimensional linear control systems with an observer / A. A. Kozlov, A. D. Burak // Bulletin of Udmurt University. Mathematics. Mechanics. Computer Science. – 2018. – Vol. 28, No. 4. – P. 445-461. – DOI 10.20537/vm180402. – EDN YVNNED.
89. Козлов, А. А. Критерий равномерной глобальной достижимости линейных систем / А. А. Козлов // Известия Института математики и информатики Удмуртского государственного университета. – 2018. – Т. 52. – С. 47-58. – DOI 10.20537/2226-3594-2018-52-04. – EDN VNGGDK.
90. Ким Т.Ю., Прокопович Г.А. Технология создания цифровых двойников для оптимизации конструктивных параметров роботов и их систем управления. Цифровая трансформация. 2025;31(3):43-53. <https://doi.org/10.35596/1729-7648-2025-31-3-43-53>
91. Ким Т.Ю., Прокопович Г.А. Разработка метода подражательного обучения для нейросетевой системы управления движением мобильного робота на примере задачи поиска выхода из лабиринта. Информатика. 2024;21(3):48-62. <https://doi.org/10.37661/1816-0301-2024-21-3-48-62>
92. Ким, Т. Ю., Печковкий, Е. И., Печковская, А. В. (2024). Разработка целевой функции генетического алгоритма для оптимизации массы деталей редуктора при FDM-печати. Вестник Полоцкого государственного университета. Серия С. Фундаментальные науки, (2), 16-23. <https://doi.org/10.52928/2070-1624-2024-43-2-16-23>
93. Ким Т.Ю., Прокопович Г.А., Лобатый А.А. Форсированное управление движением мобильного робота. Информатика. 2022;19(3):86-100. <https://doi.org/10.37661/1816-0301-2022-19-3-86-100>
94. Burova, I. G. The Hermite-Birkhoff Problem and Local Spline Approximation / I. G. Burova // WSEAS Transactions on Mathematics. – 2024. – Vol. 23. – P. 591-598. – DOI 10.37394/23206.2024.23.62. – EDN YWMNSQ.
95. Burova, I. G. Local Splines and the Least Squares Method / I. G. Burova // WSEAS Transactions on Systems. – 2024. – Vol. 23. – P. 188-195. – DOI 10.37394/23202.2024.23.21. – EDN FESUZI.
96. Burova, I. G. The Application of Splines of the Seventh Order Approximation to the Solution of Integral Fredholm Equations / I. G. Burova, G. O. Alcybeev // WSEAS Transactions on Mathematics. – 2023. – Vol. 22. – P. 409-418. – DOI 10.37394/23206.2023.22.48. – EDN HAHGIP.

97. Burova, I. G. Application of Splines of the Seventh Order Approximation to the Solution of the Fredholm Integral Equations with Weekly Singularity / I. G. Burova, G. O. Alcybeev // WSEAS Transactions on Mathematics. – 2023. – Vol. 22. – P. 475-482. – DOI 10.37394/23206.2023.22.53. – EDN VMQDDU.
98. Dem'yanovich, Yu. K. A Posteriori Improvement in Projection Method / Yu. K. Dem'yanovich, I. G. Burova // WSEAS Transactions on Mathematics. – 2023. – Vol. 22. – P. 544-552. – DOI 10.37394/23206.2023.22.60. – EDN KRRXUI.
99. Burova, I. G. Splines of the Second and Seventh Order Approximation and the Stability of the Solution of the Fredholm Integral Equations of the Second Kind / I. G. Burova, G. O. Alcybeev, S. A. Schiptcova // WSEAS Transactions on Mathematics. – 2023. – Vol. 23. – P. 1-15. – DOI 10.37394/23206.2024.23.1. – EDN DSEYAK.
100. Ryabov, V. M. About the Tikhonov Regularization Method for the Solution of Incorrect Problems / V. M. Ryabov, I. G. Burova // WSEAS Transactions on Systems. – 2023. – Vol. 22. – P. 656-662. – DOI 10.37394/23202.2023.22.66. – EDN APYWXU.
101. Алцыбеев, Г. О. О применении сплайнов к решению интегральных уравнений со слабой особенностью / Г. О. Алцыбеев, И. Г. Бурова // Процессы управления и устойчивость. – 2023. – Т. 10, № 1. – С. 105-109. – EDN SKVWDU.
102. Burova, I. G. Fredholm Integral Equation and Splines of the Fifth Order of Approximation / I. G. Burova // WSEAS Transactions on Mathematics. – 2022. – Vol. 21. – P. 260-270. – DOI 10.37394/23206.2022.21.31. – EDN KCFJCS.
103. Burova, I. G. Nonlinear Integro-differential Equations and Splines of the Fifth Order of Approximation / I. G. Burova, Yu. K. Demyanovich // WSEAS Transactions on Mathematics. – 2022. – Vol. 21. – P. 691-700. – DOI 10.37394/23206.2022.21.81. – EDN UZMOLQ.
104. Burova, I. G. The Local Nonpolynomial Splines and Solution of Integro-Differential Equations / I. G. Burova // WSEAS Transactions on Mathematics. – 2022. – Vol. 21. – P. 718-730. – DOI 10.37394/23206.2022.21.84. – EDN DJBDZC.
105. Модифицированный метод Бубнова-Галеркина для решения краевых задач с линейным обыкновенным дифференциальным уравнением / Н. К. Волосова, К. А. Волосов, А. К. Волосова [и др.] // Computational Mathematics and Information Technologies. – 2024. – Т. 8, № 3. – С. 23-33. – DOI 10.23947/2587-8999-2024-8-3-23-33. – EDN UZLHKP.
106. Бурова, И. Г. Применение сплайновых аппроксимаций второго порядка к решению интегральных уравнений второго рода / И. Г. Бурова, Г. О. Алцыбеев // Вычислительные методы и программирование. – 2025. – Т. 26, № 2. – С. 175-191. – DOI 10.26089/NumMet.v26r213. – EDN OTOUPI.
107. Математические модели и методы прогнозирования процессов биологической кинетики с учетом влияния кислородного режима / А. И. Сухинов, Ю. В. Белова, И. Ю. Кузнецова [и др.] // Вестник Южно-Уральского государственного университета. Серия: Математическое моделирование и программирование. – 2025. – Т. 18, № 2. – С. 52-65. – DOI 10.14529/mmp250205. – EDN NLQJNJ.
108. Сухинов, А. И. Идентификация планктонных популяций на поверхности морских систем на основе методов машинного обучения / А. И. Сухинов, В. В. Сидорякина, Д. А. Соломаха // Приоритетные направления развития науки и образования в условиях формирования технологического суверенитета : Материалы международной научно-практической конференции, Донской государственный

- технический университет, 08–09 августа 2024 года. – Ростов-на-Дону: Донской государственный технический университет, 2024. – С. 272-277. – EDN BLQXXM.
109. Сухинов, А. И. Математическое моделирование катастрофических сгонно-нагонных явлений в Азовском море на основе данных дистанционного зондирования Земли / А. И. Сухинов, С. В. Проценко, Н. Д. Панасенко // Интеллектуальные информационные технологии и математическое моделирование : труды Международной научно-практической конференции ИИТ&ММ-2024, пос. Дивноморское, г. Геленджик, 24–30 августа 2024 года. – Ростов-на-Дону: Общество с ограниченной ответственностью "ДГТУ-ПРИНТ", 2024. – С. 62-67. – EDN HJYFLK.
110. Аналитические и численные аспекты решения задачи биологической кинетики в присутствии частиц микропластика / А. И. Сухинов, А. Е. Чистяков, Ю. В. Белова, И. Ю. Кузнецова // Актуальные проблемы прикладной математики, информатики и механики : сборник трудов Международной научной конференции, Воронеж, 04–06 декабря 2023 года. – Воронеж: Общество с ограниченной ответственностью "Вэлборн", Издательство «Научно-исследовательские публикации», 2024. – С. 695-699. – EDN EOPFSK.
111. Sukhinov, A. I. Comparative Analysis of Numerical and Analytical Studies of Hydrodynamic Processes in Shallow Water Bodies / A. I. Sukhinov, E. A. Protsenko, S. V. Protsenko // *Water Resources*. – 2024. – Vol. 51, No. S2. – P. S216-S230. – DOI 10.1134/S0097807824701525. – EDN BPDTXZ.
112. Аналитическое и численное исследование задачи динамики планктонных популяций при наличии микропластика / А. И. Сухинов, А. Е. Чистяков, Ю. В. Белова, И. Ю. Кузнецова // Математическое моделирование. – 2024. – Т. 36, № 3. – С. 95-114. – DOI 10.20948/mm-2024-03-07. – EDN PMMLHV.
113. Использование параллельных вычислений для оценки процесса переноса загрязняющих веществ в мелководных водоемах / А. И. Сухинов, А. Е. Чистяков, В. В. Сидорякина [и др.] // Известия Саратовского университета. Новая серия. Серия: Математика. Механика. Информатика. – 2024. – Т. 24, № 2. – С. 298-315. – DOI 10.18500/1816-9791-2024-24-2-298-315. – EDN LXBVIR.
114. Проценко, Е. А. Численное моделирование гидродинамических волновых процессов в Азовском море на основе ветроволновой модели WAVEWATCH III / Е. А. Проценко, А. И. Сухинов, С. В. Проценко // Вычислительная механика сплошных сред. – 2024. – Т. 17, № 4. – С. 422-431. – DOI 10.7242/1999-6691/2024.17.4.34. – EDN AQJUTT.
115. Parallel Numerical Implementation of Mathematical Wave Hydrodynamics Models Taking into Account the Features of the Vertical Turbulent Exchange Using Remote Sensing Data / A. I. Sukhinov, E. A. Protsenko, S. V. Protsenko, N. D. Panasenko // *Mathematical Models and Computer Simulations*. – 2024. – Vol. 16, No. 2. – P. 267-279. – DOI 10.1134/S2070048224020170. – EDN UJXEWX.
116. Analytical and Numerical Study of the Problem of the Plankton Population Dynamics in the Presence of Microplastics / A. I. Sukhinov, A. E. Chistyakov, Yu. V. Belova, I. Yu. Kuznetsova // *Mathematical Models and Computer Simulations*. – 2024. – Vol. 16, No. 5. – P. 717-729. – DOI 10.1134/S2070048224700352. – EDN EYBSKL.
117. Sukhinov, A. I. WAVEWATCH III Hybrid Parallelization for Azov Sea Wave Modelling / A. I. Sukhinov, E. A. Protsenko, S. V. Protsenko // *Supercomputing Frontiers*

- and Innovations. – 2024. – Vol. 11, No. 1. – P. 81-96. – DOI 10.14529/jsfi240104. – EDN TFAPTZ.
118. Sukhinov, A. Modeling the trajectories of floating and non-floating microplastic particles in the water column / A. Sukhinov, E. Protsenko, S. Protsenko // E3S Web of Conferences. – 2024. – Vol. 583. – P. 03024. – DOI 10.1051/e3sconf/202458303024. – EDN HVSMOG.
119. Panasenko, N. Forecasting coastal systems based on satellite images / N. Panasenko, A. Sukhinov // E3S Web of Conferences. – 2024. – Vol. 592. – P. 06022. – DOI 10.1051/e3sconf/202459206022. – EDN CYWUQV.
120. Сухинов, А. И. Математическое моделирование пространственно-неоднородного нестационарного взаимодействия вредителей с трансгенной и немодифицированной агрокультурами с учетом таксиса / А. И. Сухинов, И. А. Бугаева // Computational Mathematics and Information Technologies. – 2024. – Т. 8, № 3. – С. 9-22. – DOI 10.23947/2587-8999-2024-8-3-9-22. – EDN BVAQUB.
121. Sukhinov, A. Identification of plankton populations in the surface waters of the Azov Sea based on neural network structures of various architectures / A. Sukhinov, V. Sidoryakina, D. Solomakha // BIO Web of Conferences. – 2024. – Vol. 141. – P. 03003. – DOI 10.1051/bioconf/202414103003. – EDN RPROFC.
122. Sukhinov, A. Modeling of biogeochemical processes in the Azov Sea based on the Azov3D software package / A. Sukhinov, Yu. Belova, A. Atayan // BIO Web of Conferences. – 2024. – Vol. 145. – P. 02018. – DOI 10.1051/bioconf/202414502018. – EDN PGGLJR.
123. Sukhinov, A. I. Modeling effects of 3d turbulent currents with regular waves using the les approach based on filtered adcp data / A. I. Sukhinov, E. A. Protsenko, S. V. Protsenko // Актуальные проблемы прикладной математики, информатики и механики : сборник трудов Международной научной конференции, Воронеж, 12–14 декабря 2022 года / Воронежский государственный университет. – Воронеж: Научно-исследовательские публикации, 2023. – P. 249-255. – EDN CWVFIH.
124. Сидорякина, В. В. Построение и исследование близости решений в L2 двух краевых задач для модели переноса многокомпонентных взвесей в прибрежных системах / В. В. Сидорякина, А. И. Сухинов // Журнал вычислительной математики и математической физики. – 2023. – Т. 63, № 10. – С. 1721-1732. – DOI 10.31857/S0044466923100149. – EDN LIKHNN.
125. Параллельная численная реализация моделей волновой гидродинамики с учетом особенностей вертикального турбулентного обмена / А. И. Сухинов, А. И. Sukhinov, Е. А. Проценко [et al.] // Математическое моделирование. – 2023. – Vol. 35, No. 12. – P. 51-68. – DOI 10.20948/mm-2023-12-04. – EDN KACGXO.
126. Метод решения сеточных уравнений для задач гидродинамики в плоских областях / А. И. Сухинов, А. Е. Чистяков, А. В. Никитина [и др.] // Математическое моделирование. – 2023. – Т. 35, № 3. – С. 35-58. – DOI 10.20948/mm-2023-03-03. – EDN DVDGSG.
127. Нанопоровое секвенирование для решения задач синтетической биологии / А. В. Сергеев, В. А. Родин, Т. В. Панова [и др.] // Современные вызовы структурной и синтетической биологии : материалы всероссийской школы-конференции, Шеремет, 03–07 апреля 2024 года. – Новосибирск: Офсет-ТМ, 2024. – С. 31-32. – EDN XKUDSG.

128. Исследование перспективности использования нейронных сетей для поиска новых высокомолекулярных высокоэнергетических веществ / В. М. Волохов, Е. С. Амосова, В. В. Парахин [и др.] // Суперкомпьютерные дни в России : Труды международной конференции, Москва, 23–24 сентября 2024 года. – Москва: ООО "МАКС Пресс", 2024. – С. 75-83. – EDN BQLPSZ.
129. Квантово-химическое моделирование высокоэнергетических фтординитрометилазоксисоединений / В. М. Волохов, В. В. Парахин, Е. С. Амосова [и др.] // Вычислительные методы и программирование. – 2024. – Т. 25, № 5. – С. 82-96. – DOI 10.26089/NumMet.v24s06. – EDN FAXXUT.
130. An Approach to Solving the Problem of Supercomputer Co-design / A. S. Antonov, R. V. Maier, D. A. Nikitenko, V. V. Voevodin // Lobachevskii Journal of Mathematics. – 2024. – Vol. 45, No. 7. – P. 2965-2973. – DOI 10.1134/S1995080224603680. – EDN JHFVQW.
131. New Capabilities in Exoplanet Research Using Unique MASTER Global Robotic Net Archive with Lomonosov Supercomputer Powers / A. N. Tarasenkov, V. M. Lipunov, G. A. Antipov [et al.] // Lobachevskii Journal of Mathematics. – 2024. – Vol. 45, No. 7. – P. 3188-3197. – DOI 10.1134/S1995080224603783. – EDN CHRETX.
132. Энергетический потенциал трис(пирроло)-, трис(диазоло)бензолов и 1,3,5-азинов как модельных компонентов твердых топлив / В. В. Парахин, В. М. Волохов, Д. Б. Лемперт [и др.] // Горение и взрыв. – 2024. – Т. 17, № 3(44). – С. 111-126. – DOI 10.30826/CE24170311. – EDN GGBJPJ.
133. Quantum-Chemical Study of Some Trispyrazolobenzenes and Trispyrazolo-1,3,5-triazines / V. M. Volokhov, V. V. Parakhin, E. S. Amosova [et al.] // Supercomputing Frontiers and Innovations. – 2024. – Vol. 11, No. 3. – P. 64-73. – DOI 10.14529/jsfi240304. – EDN JDRSBH.
134. Tool and Algorithm for the Determination of Aptamers in Nanopore Sequencing Data: AptaLong / M. A. Grigoryeva, M. G. Khrenova, M. F. Subach [et al.] // Supercomputing Frontiers and Innovations. – 2024. – Vol. 11, No. 3. – P. 93-106. – DOI 10.14529/jsfi240306. – EDN FPOIPV.
135. Поздравление с юбилеем члена-корреспондента РАН А.И. Сухинова / А. П. Ч. Петров, Ю. В. Василевский, В. В. Воеводин [и др.] // Computational Mathematics and Information Technologies. – 2024. – Т. 8, № 3. – С. 7-8. – EDN LMBOPR.
136. Иванов, Д. А. Оптимизация нейронных сетей в задачах обучения с подкреплением с использованием разреженных вычислений / Д. А. Иванов, В. В. Воеводин // Ломоносовские чтения : научная конференция, Москва, 14–22 апреля 2022 года / МГУ имени М. В. Ломоносова. – Москва: ООО "МАКС Пресс", 2022. – С. 109-110. – EDN TSKVMN.
137. Guide tree optimization with genetic algorithm to improve multiple protein 3D-structure alignment / M. V. Shegay, V. V. Voevodin, N. N. Popova [et al.] // Bioinformatics. – 2022. – Vol. 38, No. 4. – P. 985-989. – DOI 10.1093/bioinformatics/btab798. – EDN LIVRWG.
138. VGL: a high-performance graph processing framework for the NEC SX-Aurora TSUBASA vector architecture / I. V. Afanasyev, V. V. Voevodin, K. Komatsu, H. Kobayashi // The Journal of Supercomputing. – 2021. – DOI 10.1007/s11227-020-03564-9. – EDN MEAPPI.

139. Антонов, А. С. Высокопроизводительные вычислительные платформы: текущий статус и тенденции развития / А. С. Антонов, И. В. Афанасьев, В. В. Воеводин // *Вычислительные методы и программирование*. – 2021. – Т. 22, № 2. – С. 135-177. – DOI 10.26089/NumMet.v22r210. – EDN VFWDZY.
140. A Review of Supercomputer Performance Monitoring Systems / K. S. Stefanov, S. Pawar, A. Ranjan [et al.] // *Supercomputing Frontiers and Innovations*. – 2021. – Vol. 8, No. 3. – P. 62-81. – DOI 10.14529/jsfi210304. – EDN FZIAVM.
141. Сравнительный анализ 3D-структур белков на GPU: ре-имплементация алгоритма МАТТ на CUDA / И. А. Тимохин, Д. А. Суплатов, Н. Н. Попова [и др.] // *Суперкомпьютерные дни в России : Труды международной конференции, Москва, 21–22 сентября 2020 года / Под. редакцией Вл.В. Воеводина*. – Москва: ООО "МАКС Пресс", 2020. – С. 147-148. – EDN EQXKAD.
142. Afanasyev, I. V. Developing Efficient Implementations of Connected Component Algorithms for NEC SX-Aurora TSUBASA / I. V. Afanasyev, Vl. V. Voevodin // *Lobachevskii Journal of Mathematics*. – 2020. – Vol. 41, No. 8. – P. 1417-1426. – DOI 10.1134/S1995080220080028. – EDN LVVLFH.
143. Antonov, A. S. Algo500-A New Approach to the Joint Analysis of Algorithms and Computers / A. S. Antonov, D. A. Nikitenko, Vl. V. Voevodin // *Lobachevskii Journal of Mathematics*. – 2020. – Vol. 41, No. 8. – P. 1435-1443. – DOI 10.1134/S1995080220080041. – EDN RUNCBV.
144. Developing an efficient vector-friendly implementation of the breadth-first search algorithm for NEC SX-aurora tsubasa / I. V. Afanasyev, V. V. Voevodin, K. Komatsu, H. Kobayashi // *Communications in Computer and Information Science*. – 2020. – Vol. 1263. – P. 131-145. – DOI 10.1007/978-3-030-55326-5_10. – EDN WFYMKT.
145. Antonov, A. S. Parallel structure of algorithms and training computational technology specialists / A. S. Antonov, V. V. Voevodin, N. N. Popova // *Journal of Physics: Conference Series, Voronezh, 08–09 февраля 2018 года*. Vol. 1202. – Voronezh: Institute of Physics Publishing, 2019. – P. 12021. – DOI 10.1088/1742-6596/1202/1/012021. – EDN GCNBEJ.
146. Воеводин, В. В. Суперкомпьютерные технологии в цифровом мире: теория, практика, образование / В. В. Воеводин // *Вестник Российской академии наук*. – 2019. – Т. 89, № 4. – С. 351-354. – DOI 10.31857/S0869-5873894351-354. – EDN RPOTJG.
147. Анализ сложных слабоформализуемых природно-технических систем при помощи технологии когнитивного моделирования / В. И. Балута, С. С. Варыханов, В. П. Осипов [и др.] // *Математическое моделирование*. – 2025. – Т. 37, № 2. – С. 111-127. – DOI 10.20948/mm-2024-02-09. – EDN TSPJNV.
148. Четверушкин, Б. Н. Использование кинетической модели для нахождения пульсационных моментов турбулентного течения / Б. Н. Четверушкин, А. Е. Луцкий, Е. В. Шильников // *Математическое моделирование*. – 2025. – Т. 37, № 2. – С. 170-184. – DOI 10.20948/mm-2025-02-13. – EDN TADQWM.
149. Выявление атак типа триггер на искусственные нейронные сети / А. А. Зацаринный, А. А. Карандеев, В. П. Осипов [и др.] // *Информационные технологии и вычислительные системы*. – 2025. – № 1. – С. 3-13. – DOI 10.14357/20718632250101. – EDN CPNGAB.
150. Четверушкин, Б. Н. Алгоритм кластеризации графа на примере исследования структуры взаимодействия предметных областей в наукометрической базе / Б. Н.

- Четверушкин, М. В. Яшина, А. Г. Таташев // XIV Всероссийское совещание по проблемам управления : сборник научных трудов, Москва, 17–20 июня 2024 года. – Москва: Институт проблем управления им. В.А. Трапезникова РАН, 2024. – С. 2062-2067. – EDN KVHVEZ.
151. Chetverushkin, B. N. A Model for Describing Turbulent Flows / B. N. Chetverushkin, A. E. Lutsky, E. V. Shilnikov // *Differential Equations*. – 2024. – Vol. 60, No. 7. – P. 976-984. – DOI 10.1134/S0012266124070115. – EDN RRZGQQ.
152. Четверушкин, Б. Н. Вычислительные алгоритмы и прикладное математическое обеспечение для систем высокой и сверхвысокой производительности. Образовательный аспект / Б. Н. Четверушкин, М. В. Якововский // *Вычислительные методы и программирование*. – 2024. – Т. 25, № 5. – С. 97-106. – DOI 10.26089/NumMet.2024s07. – EDN ZEWDPK.
153. Численное моделирование трехмерного обтекания воздухозаборника / Б. Н. Четверушкин, В. Е. Борисов, А. Е. Луцкий, Я. В. Ханхасаева // *Математическое моделирование*. – 2024. – Т. 36, № 3. – С. 51-66. – DOI 10.20948/mm-2024-03-04. – EDN TLOXAK.
154. Chetverushkin, B. N. Description of Turbulent Flows Using a Kinetic Model / B. N. Chetverushkin, A. E. Lutsky, E. V. Shilnikov // *Doklady Mathematics*. – 2024. – DOI 10.1134/s1064562424701953. – EDN CPDHFP.
155. Chetverushkin, B. N. Graph Condensation for Large Factor Models / B. N. Chetverushkin, V. A. Sudakov, Yu. P. Titov // *Doklady Mathematics*. – 2024. – DOI 10.1134/s1064562424702090. – EDN XHGGNS.
156. Chetverushkin, B. N. Stabilized Scheme for Calculating Radiation Transfer in the P1 Approximation / B. N. Chetverushkin, O. G. Olkhovskaya, V. A. Gasilov // *Doklady Mathematics*. – 2024. – DOI 10.1134/s1064562424601495. – EDN ZQMTFL.
157. Четверушкин, Б. Н. Об одной модели для описания турбулентных течений / Б. Н. Четверушкин, А. Е. Луцкий, Е. В. Шильников // *Дифференциальные уравнения*. – 2024. – Т. 60, № 7. – С. 990-1000. – DOI 10.31857/S0374064124070109. – EDN KNDPWI.
158. Numerical Simulation of a 3D Flow Past an Inlet Model / B. N. Chetverushkin, V. E. Borisov, A. E. Lutsky, Ya. V. Khankhasaeva // *Mathematical Models and Computer Simulations*. – 2024. – Vol. 16, No. 5. – P. 687-697. – DOI 10.1134/S2070048224700327. – EDN FIQHVS.
159. Четверушкин, Б. Н. Описание турбулентных течений с помощью кинетической модели / Б. Н. Четверушкин, А. Е. Луцкий, Е. В. Шильников // *Доклады Российской академии наук. Математика, информатика, процессы управления*. – 2024. – Т. 516, № 1. – С. 51-58. – DOI 10.31857/S2686954324020086. – EDN XISWSM.
160. Четверушкин, Б. Н. Конденсация графов для больших факторных моделей / Б. Н. Четверушкин, В. А. Судаков, Ю. П. Титов // *Доклады Российской академии наук. Математика, информатика, процессы управления*. – 2024. – Т. 517, № 1. – С. 66-73. – DOI 10.31857/S2686954324030119. – EDN YAUMIS.
161. Четверушкин, Б. Н. Стабилизированная схема для расчета переноса излучения в приближении P1 / Б. Н. Четверушкин, О. Г. Ольховская, В. А. Гасилов // *Доклады Российской академии наук. Математика, информатика, процессы управления*. – 2024. – Т. 519, № 1. – С. 7-13. – DOI 10.31857/S2686954324050024. – EDN XEUSHK.

162. Четверушкин, Б. Н. Кинетические модели в механике сплошной среды / Б. Н. Четверушкин // Тезисы докладов XX научно-технической конференции по аэроакустике, Суздаль, 24–29 сентября 2023 года. – Суздаль: Центральный аэрогидродинамический институт имени профессора Н.Е. Жуковского, 2023. – С. 12. – EDN CPBRWM.
163. О разработке самообучающейся системы диагностики состояния природно-технических систем / В. И. Балута, С. С. Варыханов, А. Ф. Жирков [и др.] // X международная конференция по математическому моделированию, посвященная 30-летию Академии наук Республики Саха (Якутия) и памяти первого Президента Академии наук РС(Я), член-корреспондента РАН Филиппова Василия Васильевича : Тезисы докладов, Якутск, 17–20 июля 2023 года. – Якутск: Северо-Восточный федеральный университет имени М.К. Аммосова, 2023. – С. 88-89. – EDN QYXRMO.
164. Четверушкин, Б. Н. Анализ веб-графа научной периодики для выявления тенденций развития транспортных наук / Б. Н. Четверушкин, М. В. Яшина, А. Г. Таташев // XVI Всероссийская мультikonференция по проблемам управления (МКПУ-2023) : материалы мультikonференции. В 4 т., Волгоград, 11–15 сентября 2023 года. Том 4. – Волгоград: Волгоградский государственный технический университет, 2023. – С. 105-108. – EDN GFOEDU.
165. Chetverushkin, B. Kinetic consistent MHD algorithm for incompressible conductive fluids / B. Chetverushkin, A. Saveliev, V. Saveliev // *Computers & Fluids*. – 2023. – Vol. 255. – P. 105724. – DOI 10.1016/j.compfluid.2022.105724. – EDN LGGJYW.
166. Almohamed, M. Specific Cases of One General Inverse Problem for Abstract Differential Equations / M. Almohamed, I. V. Tikhonov // *Lobachevskii Journal of Mathematics*. – 2023. – Vol. 44, No. 2. – P. 502-509. – DOI 10.1134/S1995080223020063. – EDN WWHIZW.
167. Tikhonov, I. V. Generalized Popoviciu Expansions for Bernstein Polynomials of a Rational Module / I. V. Tikhonov, V. B. Sherstyukov, D. G. Tsvetkovich // *Journal of Mathematical Sciences*. – 2022. – Vol. 263, No. 4. – P. 532-578. – DOI 10.1007/s10958-022-05948-z. – EDN AUJLUL.
168. Chistyakov, A. Ev. Mathematical modeling of oscillatory processes with a free boundary / A. Ev. Chistyakov, E. An. Protsenko, E. F. Timofeeva // *Computational Mathematics and Information Technologies*. – 2017. – Vol. 1, No. 1. – P. 102-112. – DOI 10.23947/2587-8999-2017-1-102-112. – EDN YTKOKT.
169. Nikitina, A. V. Mathematical modeling of eutrophication processes in Azov Sea on supercomputers / A. V. Nikitina, A. A. Semenyakina // *Computational Mathematics and Information Technologies*. – 2017. – Vol. 1, No. 1. – P. 82-101. – DOI 10.23947/2587-8999-2017-1-82-101. – EDN YTKOKK.
170. Shcherbakova, E. E. Solving the eigenvalues and eigenfunctions problems for the Helmholtz equation by the point-sources method / E. E. Shcherbakova // *Computational Mathematics and Information Technologies*. – 2017. – Vol. 1, No. 1. – P. 54-64. – DOI 10.23947/2587-8999-2017-1-54-64. – EDN YTKOJJ.
171. Mathematical modelling of thermoelastic behavior of a coating taking into account frictional heating and wear / V. B. Zelentsov, B. I. Mitrin, I. A. Lubyagin, S. M. Aizikovich // *Computational Mathematics and Information Technologies*. – 2017. – Vol. 1, No. 1. – P. 36-53. – DOI 10.23947/2587-8999-2017-1-36-53. – EDN YTKOJA.

172. Sukhinov, A. I. Sufficient convergence conditions for positive solutions of linearized two-dimensional sediment transport problem / A. I. Sukhinov, V. V. Sidoryakina, A. A. Sukhinov // *Computational Mathematics and Information Technologies*. – 2017. – Vol. 1, No. 1. – P. 21-35. – DOI 10.23947/2587-8999-2017-1-21-35. – EDN YTKOIR.
173. Petrov, I. B. Application of grid-characteristic method for numerical solution of deformable solid mechanics dynamical problems / I. B. Petrov // *Computational Mathematics and Information Technologies*. – 2017. – Vol. 1, No. 1. – P. 1-20. – DOI 10.23947/2587-8999-2017-1-1-20. – EDN YTKOII.
174. Afendikov, A. L. On Cartesian grids in some adaptive algorithms of aerodynamics / A. L. Afendikov, V. S. Nikitin // *Computational Mathematics and Information Technologies*. – 2017. – Vol. 1, No. 2. – P. 54-58. – DOI 10.23947/2587-8999-2017-2-180-184. – EDN YTKONW.
175. Karamzin, Yu. N. On a class of flux schemes for convection-diffusion equations / Yu. N. Karamzin, T. A. Kudryashova, S. V. Polyakov // *Computational Mathematics and Information Technologies*. – 2017. – Vol. 1, No. 2. – P. 42-53. – DOI 10.23947/2587-8999-2017-2-169-179. – EDN YTKONN.
176. Shutyaev, V. P. Study of the optimal solution sensitivity of variational data assimilation problem for a model of sea thermodynamics / V. P. Shutyaev, E. I. Parmuzin // *Computational Mathematics and Information Technologies*. – 2017. – Vol. 1, No. 2. – P. 29-35. – DOI 10.23947/2587-8999-2017-2-156-163.
177. Efficient parallel software system for solving Navier-Stokes equations by the discontinuous Galerkin method / M. M. Krasnov, P. A. Kuchugov, M. E. Ladonkina, V. F. Tishkin // *Computational Mathematics and Information Technologies*. – 2017. – Vol. 1, No. 2. – P. 21-28. – DOI 10.23947/2587-8999-2017-2-148-155. – EDN YTKOMM.
178. Agoshkov, V. I. New approaches to the formulation of domain decomposition method and algorithm of the large-block parallelization for mathematical modeling problems / V. I. Agoshkov, N. R. Lezina // *Computational Mathematics and Information Technologies*. – 2017. – Vol. 1, No. 2. – P. 14-20. – DOI 10.23947/2587-8999-2017-2-141-147. – EDN YTKOLU.
179. Sukhinov, A. I. Using multichannel satellite images for predictive modelling the "bloom" phytoplankton processes in shallow waters on supercomputer / A. I. Sukhinov, A. V. Nikitina, A. E. Chistyakov // *Computational Mathematics and Information Technologies*. – 2017. – Vol. 1, No. 2. – P. 1-13. – DOI 10.23947/2587-8999-2017-2-128-140. – EDN NTLQRX.
180. Сухинов, А. И. Адаптивный модифицированный попеременно-треугольный метод / А. И. Сухинов, А. Е. Чистяков // *Computational Mathematics and Information Technologies*. – 2018. – Т. 2, № 1. – С. 44-59. – DOI 10.23947/2587-8999-2018-2-1-44-59. – EDN VSNPTK.
181. Никитина, А. В. Математическое моделирование плотности морской воды глубоководного водоема / А. В. Никитина, А. Л. Леонтьев // *Computational Mathematics and Information Technologies*. – 2018. – Т. 2, № 2. – С. 121-132. – DOI 10.23947/2587-8999-2018-2-2-121-132. – EDN YVEKWL.
182. Математическая модель газодинамических и тепловых процессов в паровой турбине / А. И. Сухинов, А. Е. Чистяков, Н. Н. Ефимов, В. Н. Балтян // *Computational Mathematics and Information Technologies*. – 2018. – Т. 2, № 2. – С. 106-120. – DOI 10.23947/2587-8999-2018-2-2-106-120. – EDN VSNPZW.

183. Моделирование прибрежной аэродинамики с учетом лесных насаждений / Ю. В. Белова, Е. А. Проценко, А. М. Атаян, И. А. Курская // *Computational Mathematics and Information Technologies*. – 2018. – Т. 2, № 2. – С. 91-105. – DOI 10.23947/2587-8999-2018-2-2-91-105. – EDN YVEKVN.
184. Сухинов, А. И. Построение и исследование корректности математической модели транспорта и осаждения взвесей с учетом изменений рельефа дна / А. И. Сухинов, В. В. Сидорякина // *Computational Mathematics and Information Technologies*. – 2018. – Т. 2, № 2. – С. 76-90. – DOI 10.23947/2587-8999-2018-2-76-90. – EDN YVEKSL.
185. Семенякина, А. А. Разработка модели транспорта и трансформации соединений азота, фосфора и кремния в мелководном водоеме / А. А. Семенякина, В. В. Сумбаев, С. В. Проценко // *Computational Mathematics and Information Technologies*. – 2018. – Т. 2, № 2. – С. 67-75. – DOI 10.23947/2587-8999-2018-2-67-75. – EDN YKVKYO.
186. Improvement of numerical solution smoothness for the hydrodynamics problems modeling on rectangular grids / A. I. Sukhinov, A. E. Chistyakov, E. A. Protsenko [et al.] // *Computational Mathematics and Information Technologies*. – 2019. – Vol. 1, No. 1. – P. 1-16. – DOI 10.23947/2587-8999-2019-1-1-1-16. – EDN BOYBYI.
187. Chistyakov, A. E. Linear combination of the Upwind and Standard Leapfrog difference schemes with weight coefficients obtained by minimizing the approximation error / A. E. Chistyakov, A. M. Atayan // *Computational Mathematics and Information Technologies*. – 2019. – Vol. 2, No. 2. – P. 54-66. – DOI 10.23947/2587-8999-2019-1-1-1-16. – EDN WPGKXV.
188. Mathematical modeling of nonlinear effects in dynamic of interacting plankton and fish populations of Azov Sea / A. I. Sukhinov, V. V. Sidoryakina, A. V. Nikitina [et al.] // *Computational Mathematics and Information Technologies*. – 2019. – Vol. 2, No. 2. – P. 83-103. – DOI 10.23947/2587-8999-2019-2-2-83-103. – EDN PPZFMО.
189. Modified Upwind Leapfrog difference scheme / A. I. Sukhinov, A. E. Chistyakov, I. Y. Kuznetsova [et al.] // *Computational Mathematics and Information Technologies*. – 2020. – Vol. 1, No. 1. – P. 56-70. – EDN CPIJTG.
190. Sukhinov, A. I. Two-dimensional splitting schemes for hyperbolic equations / A. I. Sukhinov // *Computational Mathematics and Information Technologies*. – 2020. – Vol. 1, No. 2. – P. 71-86. – DOI 10.23947/2587-8999-2020-1-2-71-86. – EDN ZUPYQA.
191. Sukhinov, A. I. Suspensions transport predictive modeling under various scenarios of ferromanganese nodules extraction from the Pacific ocean bottom / A. I. Sukhinov, A. A. Sukhinov, S. V. Kirilchik // *Computational Mathematics and Information Technologies*. – 2021. – Vol. 1, No. 1. – P. 1-9. – EDN SQORFY.
192. Simulation of microbiological destruction process of oil pollution in shallow water / A. A. Filina, V. V. Sidoryakina, A. E. Chistyakov, A. V. Nikitina // *Computational Mathematics and Information Technologies*. – 2021. – Vol. 1, No. 1. – P. 10-25. – EDN UJHSEC.
193. Protsenko, S. V. Vertical turbulent exchange coefficient parametrization in the framework of the LES approach / S. V. Protsenko, A. M. Atayan // *Computational Mathematics and Information Technologies*. – 2021. – Vol. 1, No. 1. – P. 26-35. – EDN DDRWSW.
194. Sukhinov, A. I. Hydrodynamic wave processes numerical modeling in the coastal recreational zone of the Tsimlyansk reservoir / A. I. Sukhinov, V. V. Sidoryakina, E. A.

- Protsenko // Computational Mathematics and Information Technologies. – 2021. – Vol. 1, No. 1. – P. 36-43. – EDN CTRRCP.
195. Shvedovskiy, V. A. Cellular automaton with percolation as a dynamic system: entropy approach / V. A. Shvedovskiy // Computational Mathematics and Information Technologies. – 2021. – Vol. 1, No. 2. – P. 61-71. – EDN JOUEOX.
196. Sidoryakina, V. V. Various approximations of vertical turbulent exchange parameterization for the analysis of the waves hydrodynamic impact on the bottom of the reservoir / V. V. Sidoryakina, S. V. Protsenko // Computational Mathematics and Information Technologies. – 2021. – Vol. 1, No. 2. – P. 78-85. – EDN ISPNIR.
197. In memory of Alexander Petrovich Mikhailov / B. N. Chetverushkin, V. F. Tishkin, M. V. Yakobovsky [et al.] // Computational Mathematics and Information Technologies. – 2022. – Vol. 1, No. 1. – P. 1-11. – DOI 10.23947/2587-8999-2022-1-1-1-11. – EDN JQPBHH.
198. Khokhlov, N. I. Application of the grid-characteristic method for solving the problems of the propagation of dynamic waves using HPC systems / N. I. Khokhlov // Computational Mathematics and Information Technologies. – 2022. – Vol. 1, No. 1. – P. 20-30. – DOI 10.23947/2587-8999-2022-1-1-20-30. – EDN GZXDXX.
199. Calculation of wave hydrophysical characteristics for inland freshwater reservoirs of Southern Russia / A. I. Sukhinov, V. V. Sidoryakina, E. A. Protsenko, S. V. Protsenko // Computational Mathematics and Information Technologies. – 2022. – Vol. 1, No. 1. – P. 31-40. – DOI 10.23947/2587-8999-2022-1-1-31-40. – EDN WEWQKS.
200. Shaimardanova, G. F. Genetic algorithm for solving the inverse problem of chemical kinetics / G. F. Shaimardanova, K. F. Koledina // Computational Mathematics and Information Technologies. – 2022. – Vol. 1, No. 1. – P. 41-49. – DOI 10.23947/2587-8999-2022-1-1-41-49. – EDN IUQOSU.
201. Применение быстрых разложений для построения точных решений задачи о прогибе прямоугольной мембраны под действием переменной нагрузки/ А. Д. Чернышев, В. В. Горяйнов, С. В. Кузнецов, О. Ю. Никифорова//Вестник Томского государственного университета. Математика и механика. – 2021. №70. – С. 127-142. DOI 10.17223/19988621/70/11.
202. Некоторые точные решения задачи о прогибах прямоугольной мембраны, полученные методом быстрых разложений / А. Д. Чернышов, В. В. Горяйнов, О. Ю. Никифорова, М. И. Попов // Итоги науки : Научные исследования. – Москва : Российская академия наук, 2024. – С. 28-51. – EDN CESQDP.
203. Чернышов, А. Д. О деформировании прямоугольной пластины на упругом основании с произвольными граничными условиями / А. Д. Чернышов, С. Ф. Кузнецов, О. Ю. Никифорова // Материалы LXII отчетной научной конференции преподавателей и научных сотрудников ВГУИТ за 2023 год, Воронеж, 08–09 февраля 2024 года. – Воронеж: Воронежский государственный университет инженерных технологий, 2024. – С. 120-123. – EDN САОКYY.
204. Соответствие качества и структуры образования текущим и перспективным потребностям рынка труда / А. Д. Чернышов, О. Ю. Никифорова, С. Ф. Кузнецов [и др.] // Проблемы практической подготовки студентов : материалы XXI Всероссийской научно-практической конференции, Воронеж, 17 декабря 2024 года. – Воронеж, 2024. – С. 13-18. – EDN OHIMXL.

205. Об особенностях построения аналитического решения бигармонического уравнения методом быстрых разложений / А. Д. Чернышов, О. Ю. Никифорова, В. В. Горяйнов, С. Ф. Кузнецов // Актуальные проблемы прикладной математики, информатики и механики : Сборник трудов Международной научной конференции, Воронеж, 02–04 декабря 2024 года. – Воронеж: Научно-исследовательские публикации, 2025. – С. 1187-1190. – EDN НТРJVS.
206. О напряжениях в прямоугольной мембране, закрепленной по всей длине одной стороны и в углах противоположной стороны / В. В. Горяйнов, С. Ф. Кузнецов, О. Ю. Никифорова, И. Г. Рукин // Актуальные проблемы прикладной математики, информатики и механики : сборник трудов Международной научной конференции, Воронеж, 04–06 декабря 2023 года. – Воронеж: Общество с ограниченной ответственностью "Вэлборн", Издательство «Научно-исследовательские публикации», 2024. – С. 1061-1067. – EDN MLRBXQ.
207. Application of Universal Fast Trigonometric Interpolation and Extrapolation for Determining the Launch Point Coordinates of a Flight Vehicle / A. D. Chernyshov, O. Yu. Nikiforova, V. V. Goryainov, I. G. Rukin // Computational Mathematics and Mathematical Physics. – 2024. – Vol. 64, No. 7. – P. 1461-1471. – DOI 10.1134/S0965542524700702. – EDN EXQPKP.
208. Сидлер, И. В. О численном решении начально-краевой задачи для гиперболической системы в модели гидравлического удара / И. В. Сидлер // Динамические системы и компьютерные науки: теория и приложения (DYSC 2023) : материалы 5-й Международной конференции, Иркутск, 18–23 сентября 2023 года. – Иркутск: Иркутский государственный университет, 2023. – С. 209-212. – EDN UDJGX.
209. Алмохамед, М. Специфика одной обратной задачи с финальным переопределением второго рода / М. Алмохамед, И. В. Тихонов, Е. Д. Фуглаева // Современные проблемы математики и математического образования : Международная научная конференция «78 Герценовские чтения», Санкт-Петербург, 15–17 апреля 2025 года. – Санкт-Петербург: Издательско-полиграфическая ассоциация высших учебных заведений, 2025. – С. 280-286. – EDN LLQIZG.
210. Алмохамед, М. О кратных нулях одной целой функции, важной для теории обратных задач / М. Алмохамед, И. В. Тихонов, В. Б. Шерстюков // Владикавказский математический журнал. – 2025. – Т. 27, № 1. – С. 5-20. – DOI 10.46698/x2987-6171-9353-j. – EDN ULAEVO.
211. Тихонов, И. В. Полиномы Бернштейна на симметричном отрезке и абсолютная сходимость тейлоровских разложений / И. В. Тихонов, В. Б. Шерстюков, М. А. Петросова // Современные проблемы математики и математического образования : Международная научная конференция «78 Герценовские чтения», Санкт-Петербург, 15–17 апреля 2025 года. – Санкт-Петербург: Издательско-полиграфическая ассоциация высших учебных заведений, 2025. – С. 312-317. – EDN LSIOMX.
212. Эберт, К. А. Модели перколяции и джемминга неточечных объектов на квадратной решетке / К. А. Эберт, М. М. Бузмакова. – Пермь : Пермский государственный национальный исследовательский университет, 2024. – 86 с. – ISBN 978-5-7944-4141-3. – EDN GSASHI.
213. Эберт, К. А. Максимальная случайная упаковка блоков на кубической решетке / К. А. Эберт, М. М. Бузмакова, С. В. Русаков // Вестник Южно-Уральского государственного университета. Серия: Математическое моделирование и

- программирование. – 2024. – Т. 17, № 4. – С. 82-93. – DOI 10.14529/mmp240407. – EDN TYYVGV.
214. Бузмакова, М. М. Моделирование распространения сетевого вируса в локальной компьютерной сети методами теории перколяции / М. М. Бузмакова, Е. А. Воробьев // Вестник Пермского университета. Математика. Механика. Информатика. – 2024. – № 2(65). – С. 54-60. – DOI 10.17072/1993-0550-2024-2-54-60. – EDN GLAQRF.
215. Исследование свойств композитного материала на основе эпоксидной смолы и полидисперсной смеси кварцевого песка / М. М. Бузмакова, В. Г. Гилев, А. Ф. Мерзляков, С. В. Русаков // Вестник Пермского национального исследовательского политехнического университета. Механика. – 2024. – № 6. – С. 15-24. – DOI 10.15593/perm.mech/2024.6.02. – EDN HHQUDI.
216. Эберт, К. А. Джемминг блоков разного размера на квадратной решетке / К. А. Эберт, М. М. Бузмакова // Математическое моделирование в естественных науках. – 2024. – Т. 1. – С. 449-452. – EDN ELRXFB.
217. Бузмакова, М. М. Моделирование структуры полимерных полидисперсных сред методами теории перколяции / М. М. Бузмакова, С. В. Русаков, К. А. Эберт // Материалы XXIII Международной конференции по вычислительной механике и современным прикладным программным системам (ВМСППС'2023) : Материалы конференции, с. Дивноморское, Краснодарский край, 04–10 сентября 2023 года. – Москва: Московский авиационный институт (национальный исследовательский университет), 2023. – С. 26-28. – EDN XGKAIK.
218. Вдовин, И. Е. Компьютерное моделирование случайной плотной упаковки несжимаемых окружностей на плоскости / И. Е. Вдовин, М. М. Бузмакова // Материалы XXIII Международной конференции по вычислительной механике и современным прикладным программным системам (ВМСППС'2023) : Материалы конференции, с. Дивноморское, Краснодарский край, 04–10 сентября 2023 года. – Москва: Московский авиационный институт (национальный исследовательский университет), 2023. – С. 34-36. – EDN XWGOIH.
219. Бузмакова, М. М. Применение технологий параллельных вычислений для решения перколяционной задачи k-меров в континууме / М. М. Бузмакова, П. Д. Пучинина, С. В. Русаков // Математическое моделирование и суперкомпьютерные технологии : Труды XXIII Международной конференции, Нижний Новгород, 13–16 ноября 2023 года. – Нижний Новгород: Национальный исследовательский Нижегородский государственный университет им. Н.И. Лобачевского, 2023. – С. 37-40. – EDN HHWZLC.
220. Нестеров, А. А. Параллельные вычисления перколяционных задач на двумерной решетке / А. А. Нестеров, М. М. Бузмакова // Материалы XXIII Международной конференции по вычислительной механике и современным прикладным программным системам (ВМСППС'2023) : Материалы конференции, с. Дивноморское, Краснодарский край, 04–10 сентября 2023 года. – Москва: Московский авиационный институт (национальный исследовательский университет), 2023. – С. 70-71. – EDN PSMSTS.
221. Physical Properties of an Epoxy Composite Modified by C60 Fullerenes / М. М. Buzmakova, V. G. Gilev, A. F. Merzlyakov, S. V. Rusakov // Mechanics of Composite

- Materials. – 2018. – Vol. 54, No. 4. – P. 545-552. – DOI 10.1007/s11029-018-9763-3. – EDN VBHYDF.
222. Романова, М. П. Алгоритм диспергирования углеродных нанотрубок в тонкой пленке полимера / М. П. Романова, М. М. Бузмакова // Математическое моделирование в естественных науках. – 2018. – Т. 1. – С. 260-264. – EDN VJMQSH.
223. Бузмакова, М. М. Моделирование перколяционной системы k-меров на плоскости / М. М. Бузмакова, П. Д. Теплых // Математика и междисциплинарные исследования - 2021 : Материалы Всероссийской научно-практической конференции молодых ученых с международным участием, Пермь, 18–20 октября 2021 года / Гл. редактор А.П. Шкарапута. – Пермь: Пермский государственный национальный исследовательский университет, 2021. – С. 40-43. – EDN ZPLNGQ.
224. Романова, М. П. Модель структуры тонкой пленки эпоксидной смолы, модифицированной углеродными нанотрубками, с учетом наличия Ван-дер-Ваальсова взаимодействия / М. П. Романова, М. М. Бузмакова // Актуальные проблемы математики, механики и информатики : сборник статей по материалам студенческой конференции, Пермь, 25 мая – 10 2021 года. – Пермь: Пермский государственный национальный исследовательский университет, 2021. – С. 52-56. – EDN BCPMRU.
225. Романова, М. П. Модель структуры тонкой пленки эпоксидной смолы, модифицированной углеродными нанотрубками, с учетом наличия межфазных областей / М. П. Романова, М. М. Бузмакова // Математика и междисциплинарные исследования - 2020: Материалы Всероссийской научно-практической конференции молодых ученых с международным участием, Пермь, 12–14 октября 2020 года. – Пермь: Пермский государственный национальный исследовательский университет, 2020. – С. 295-300. – EDN MCRWFU.
226. Теплых, П. Д. Континуальная перколяция k-меров на плоскости / П. Д. Теплых, М. М. Бузмакова // Математика и междисциплинарные исследования - 2020 : Материалы Всероссийской научно-практической конференции молодых ученых с международным участием, Пермь, 12–14 октября 2020 года. – Пермь: Пермский государственный национальный исследовательский университет, 2020. – С. 301-307. – EDN ENFKJJ.
227. Бузмакова, М. М. Компьютерное моделирование джемминга для перколяционной задачи Накамуры / М. М. Бузмакова, К. А. Боков // Математические методы и информационно-технические средства : Материалы XV Всероссийской научно-практической конференции, Краснодар, 21 июня 2019 года / Редколлегия: И.Н. Старостенко [и др.]. – Краснодар: Федеральное государственное казенное образовательное учреждение высшего профессионального образования "Краснодарский университет Министерства внутренних дел Российской Федерации", 2019. – С. 27-31. – EDN NYPZEF.
228. Никитин, И. В. Моделирование структуры полимера, модифицированного углеродными нанотрубками / И. В. Никитин, М. М. Бузмакова // Математика и междисциплинарные исследования - 2019: Материалы Всероссийской научно-практической конференции молодых ученых с международным участием, Пермь, 15–18 мая 2019 года / гл. ред. А. П. Шкарапута. – Пермь: Пермский государственный национальный исследовательский университет, 2019. – С. 281-285. – EDN GCXULN.
229. Решетникова, Е. А. Алгоритм равномерного распределения k-меров в пространстве / Е. А. Решетникова, М. М. Бузмакова // Математика и междисциплинарные исследования - 2019 : Материалы Всероссийской научно-практической конференции молодых ученых с международным участием, Пермь, 15–18 мая 2019 года / гл. ред. А. П. Шкарапута. – Пермь: Пермский государственный национальный исследовательский университет, 2019. – С. 319-323. – EDN GRMXRR.
230. Романова, М. П. Континуальная модель структуры тонкой пленки полимера, модифицированной углеродными нанотрубками / М. П. Романова, М. М. Бузмакова //

- Математика и междисциплинарные исследования - 2019: Материалы Всероссийской научно-практической конференции молодых ученых с международным участием, Пермь, 15–18 мая 2019 года / гл. ред. А. П. Шкарапута. – Пермь: Пермский государственный национальный исследовательский университет, 2019. – С. 324-329. – EDN XYEDBL.
231. Семенов, В. А. Исследование анизотропной проводимости эпоксидного полимера, модифицированного углеродными нанотрубками / В. А. Семенов, С. В. Русаков, М. М. Бузмакова // Механика композиционных материалов и конструкций, сложных и гетерогенных сред: Сборник трудов 8-й Всероссийской научной конференции с международным участием им. И.Ф. Образцова и Ю.Г. Яновского, Москва, 18–19 декабря 2018 года. – Москва: Институт прикладной механики РАН, 2019. – С. 92-95. – EDN YYQLSP.
232. Бузмакова, М. М. Континуальная перколяционная модель тонкой пленки полимера, модифицированного углеродными нанотрубками / М. М. Бузмакова, М. П. Романова // Математическое и информационное моделирование: сборник научных трудов, электронный ресурс, Тюмень, 18 апреля 2019 года / Министерство науки и высшего образования Российской Федерации; Тюменский государственный университет, Институт математики и компьютерных наук. Том Выпуск 17. – Тюмень: Тюменский государственный университет, 2019. – С. 13-18. – EDN ICLIIYA.
233. Решетникова, Е. А. Сравнительный анализ алгоритмов равномерного распределения k-меров в пространстве / Е. А. Решетникова, М. М. Бузмакова // Математическое и информационное моделирование : сборник научных трудов, электронный ресурс, Тюмень, 18 апреля 2019 года / Министерство науки и высшего образования Российской Федерации; Тюменский государственный университет, Институт математики и компьютерных наук. Том Выпуск 17. – Тюмень: Тюменский государственный университет, 2019. – С. 7-12. – EDN WBMVAY.
234. Бузмакова, М. М. Экспериментальное исследование реокинетики эпоксидного связующего, модифицированного фуллеренами C60 / М. М. Бузмакова, В. Г. Гилев, С. В. Русаков // Вестник Пермского университета. Физика. – 2019. – № 2. – С. 35-40. – DOI 10.17072/1994-3598-2019-2-35-40. – EDN JMTDXL.
235. Боков, К. А. Компьютерное моделирование перколяции k-меров на квадратной решетке / К. А. Боков, М. М. Бузмакова // Вестник Пермского университета. Математика. Механика. Информатика. – 2018. – № 1(40). – С. 51-55. – DOI 10.17072/1993-0550-2018-1-51-55. – EDN XMHXWP.
236. Бузмакова, М. М. Компьютерное моделирование структуры эпоксидной смолы, модифицированной фуллеренами / М. М. Бузмакова // Морские интеллектуальные технологии. – 2017. – № 4-3(38). – С. 189-193. – EDN YPLAOG.
237. Бузмакова, М. М. Алгоритмы диспергирования частиц в неупорядоченной среде / М. М. Бузмакова, С. В. Русаков // Фундаментальные исследования. – 2016. – № 7-2. – С. 213-217. – EDN WGCFXL.
238. Агамалиева, А. И. Необходимые условия оптимальности в одной дискретной граничной задаче управления динамикой популяции / А. И. Агамалиева // Вестник Пермского университета. Математика. Механика. Информатика. – 2022. – № 2(57). – С. 5-13. – DOI 10.17072/1993-0550-2022-2-5-13. – EDN NITQOZ.
239. Гунин, В. И. Вихревая гидродинамика: новый подход к моделированию геосистем / В. И. Гунин // Вестник Пермского университета. Математика. Механика. Информатика. – 2018. – № 1(40). – С. 5-18. – DOI 10.17072/1993-0550-2018-1-5-18. – EDN XMHXRJ.
240. Чечулин, В. Л. О простых математических моделях роста культурных растений / В. Л. Чечулин // Вестник Пермского университета. Математика. Механика. Информатика. – 2018. – № 1(40). – С. 46-50. – DOI 10.17072/1993-0550-2018-1-46-50. – EDN XMHXVB.

241. Малыгина, В. В. Об устойчивости одной модели динамики популяций с последствием / В. В. Малыгина // Вестник Пермского университета. Математика. Механика. Информатика. – 2015. – № 1(28). – С. 5-10. – EDN UHSYXR.
242. Митин, В. Ю. Метод минимального покрытия и другие методы фрактального анализа изрезанности рельефа поверхностей / В. Ю. Митин // Вестник Пермского университета. Математика. Механика. Информатика. – 2013. – № 2(21). – С. 16-21. – EDN RAIOMN.
243. Садилов, Е. С. Конвекция в горизонтальном слое жидкости со свободными границами при малых числах Прандтля. Спектральный метод / Е. С. Садилов // Вестник Пермского университета. Математика. Механика. Информатика. – 2011. – № 5(9). – С. 171-174. – EDN RMRXOB.
244. Пастухов Ю.Ф. " Необходимые условия в обратной вариационной задаче ", Фундаментальная и прикладная математика,7:1(2001), 285-288.
245. Численное решение краевых задач с однородным линейным уравнением в частных производных модифицированным методом Бубнова-Галеркина на прямоугольнике / Н. К. Волосова, К. А. Волосов, А. К. Волосова [и др.] // Computational Mathematics and Information Technologies. – 2026. – Т. 10, № 1. – С. 21-36. – DOI 10.23947/2587-8999-2026-10-1-21-36. – EDN ZUJZSD.
246. Сравнение решений гидродинамической задачи в прямоугольной камере методами торможения и разгона начального поля скорости / Н. К. Волосова, К. А. Волосов, А. К. Волосова [и др.] // Computational Mathematics and Information Technologies. – 2025. – Т. 9, № 2. – С. 22-33. – DOI 10.23947/2587-8999-2025-9-2-22-33. – EDN KWAXAW.
247. Моделирование циркуляции в аневризмах кровеносных сосудов / Н. К. Волосова, К. А. Волосов, А. К. Волосова [и др.] // Computational Mathematics and Information Technologies. – 2025. – Т. 9, № 3. – С. 30-43. – DOI 10.23947/2587-8999-2025-9-3-30-43. – EDN VNQFKM.
248. Нестационарная модель свертывания крови в аневризмах кровеносных сосудов / Н. К. Волосова, К. А. Волосов, А. К. Волосова [и др.] // Computational Mathematics and Information Technologies. – 2025. – Т. 9, № 4. – С. 22-37. – DOI 10.23947/2587-8999-2025-9-4-22-37. – EDN CZQRLX.
249. Решение интегральных уравнений Фредгольма методом замены интеграла квадратурой с двенадцатым порядком погрешности в матричном виде / Н. К. Волосова, К. А. Волосов, А. К. Волосова [и др.] // Вестник Пермского университета. Математика. Механика. Информатика. – 2022. – № 4(59). – С. 9-17. – DOI 10.17072/1993-0550-2022-4-9-17. – EDN QZAGPN.
250. Обобщение метода Петрова-Галеркина для решения системы интегральных уравнений Фредгольма второго рода / Н. К. Волосова, К. А. Волосов, А. К. Волосова [и др.] // Вестник Пермского университета. Математика. Механика. Информатика. – 2023. – № 1(60). – С. 5-14. – DOI 10.17072/1993-0550-2023-1-5-14. – EDN KQEIXG.
251. Модифицированная формула Ньютона - касательных парабол на числовой оси / Н. К. Волосова, К. А. Волосов, А. К. Волосова [и др.] // Вестник Пермского университета. Математика. Механика. Информатика. – 2023. – № 2(61). – С. 5-15. – DOI 10.17072/1993-0550-2023-2-5-15. – EDN GLDLNQ.
252. Применение формул прогонки для шифрования текстовых данных / Н. К. Волосова, К. А. Волосов, А. К. Волосова [и др.] // Вестник Пермского университета.

- Математика. Механика. Информатика. – 2023. – № 3(62). – С. 5-12. – DOI 10.17072/1993-0550-2023-3-5-12. – EDN CVNPFJ.
253. N-кратное расщепление явной разностной схемы для уравнения вихря в вязкой несжимаемой жидкости / Н. К. Волосова, К. А. Волосов, А. К. Волосова [и др.] // Вестник Пермского университета. Математика. Механика. Информатика. – 2023. – № 4(63). – С. 12-21. – DOI 10.17072/1993-0550-2023-4-12-21. – EDN NYFUVJ.
254. Решение краевых задач для некоторых нелинейных дифференциальных уравнений методом Бубнова-Галеркина / Н. К. Волосова, К. А. Волосов, А. К. Волосова [и др.] // Computational Mathematics and Information Technologies. – 2025. – Т. 9, № 1. – С. 7-19. – DOI 10.23947/2587-8999-2025-9-1-7-19. – EDN IBUWGX.
255. Модифицированная формула Герасимова-Капуто / Н. К. Волосова, К. А. Волосов, А. К. Волосова [и др.] // Вестник Пермского университета. Математика. Механика. Информатика. – 2024. – № 1(64). – С. 5-14. – DOI 10.17072/1993-0550-2024-1-5-14. – EDN QLHWNH.
256. Увеличение точности решения краевых задач с линейными обыкновенными дифференциальными уравнениями методом Бубнова-Галеркина / Н. К. Волосова, К. А. Волосов, А. К. Волосова [и др.] // Computational Mathematics and Information Technologies. – 2024. – Т. 8, № 4. – С. 7-18. – DOI 10.23947/2587-8999-2024-8-4-7-18. – EDN ANIYW.
257. Chertkov, V. Information security in intelligent semantic systems / V. Chertkov // Открытые семантические технологии проектирования интеллектуальных систем. – 2022. – № 6. – Р. 417-420. – EDN DCCYUD.
258. Чертков, В. М. Метод обнаружения электронных закладных устройств на основе ам-сигнала с подавленной несущей / В. М. Чертков, В. К. Железняк // Современные средства связи. – 2020. – № 1. – С. 228-229. – EDN EDNFYE.
259. Чертков, В. М. Сравнительный анализ особенностей и форматов представления метаданных выходных продуктов систем COSMO-Skymed и Terrasar-X / В. М. Чертков, Р. П. Богущ // Современные средства связи. – 2019. – № 1. – С. 144-146. – EDN FTXKUE.
260. Чертков, В. М. Алгоритм определения меры схожести идентификационных образов закладных устройств / В. М. Чертков, В. К. Железняк // Вестник Полоцкого государственного университета. Серия С. Фундаментальные науки. – 2018. – № 4. – С. 20-27. – EDN XWQAGL.
261. Аппроксимация двойных и тройных интегралов во внутренних краевых задачах математической физики / О. В. Голубева, С. Г. Ехилевский, Н. А. Гурьева [и др.] // Вестник Полоцкого государственного университета. Серия С. Фундаментальные науки. – 2016. – № 12. – С. 86-104. – EDN XRFOMN.
262. Программное обеспечение для автоматизации профессионального отбора персонала и аттестации служащих ОАО «Нафтан» на основе модели ключевых компетенций / О. В. Голубева, В. М. Чертков, Е. Ю. Гарист, Г. Ю. Пешкова // Вестник Полоцкого государственного университета. Серия С. Фундаментальные науки. – 2024. – № 1(42). – С. 2-7. – DOI 10.52928/2070-1624-2024-42-1-2-7. – EDN AKUPVQ.
263. Пастухов, Ю. Ф. Задача построения поля линий тока по температурному разрезу / Ю. Ф. Пастухов, Д. Ф. Пастухов // Вестник Полоцкого государственного университета. Серия С. Фундаментальные науки. – 2015. – № 4. – С. 27-36. – EDN TXHZCD.

264. Матрица Гессе по старшим производным локальной записи гладкой функции в расслоении скоростей - тензор второго ранга типа $(0,2)$ / Ю. Ф. Пастухов, Н. К. Волосова, К. А. Волосов [и др.] // Тенденции развития науки и образования. – 2022. – № 85-2. – С. 28-32. – DOI 10.18411/trnio-05-2022-57. – EDN PLVSFG.
265. Матричное решение интегральных уравнений Фредгольма квадратурой с двенадцатым порядком погрешности / Н. К. Волосова, К. А. Волосов, А. К. Волосова [и др.] // Тенденции развития науки и образования. – 2022. – № 89-1. – С. 105-112. – DOI 10.18411/trnio-09-2022-31. – EDN LBUIKP.
266. Тензор Пуассона в расслоении струй гладких функций / Ю. Ф. Пастухов, Д. Ф. Пастухов, М. И. Карлов [и др.] // Тенденции развития науки и образования. – 2022. – № 89-1. – С. 116-127. – DOI 10.18411/trnio-09-2022-33. – EDN GCKVKH.
267. Пастухов, Ю. Ф. Мультидифференциальный тензор ранга S типа $(0,S)$ / Ю. Ф. Пастухов, Д. Ф. Пастухов, М. И. Карлов // Наукосфера. – 2022. – № 4-1. – С. 29-34. – EDN HPAZZK.
268. Квазилинейность в расслоенных пространствах скоростей конечного порядка - теорема о локальном представлении слоевых координат в виде функциональной квазилинейной комбинации преобразованных координат / Ю. Ф. Пастухов, Д. Ф. Пастухов, С. В. Чернов [и др.] // Тенденции развития науки и образования. – 2023. – № 95-6. – С. 124-127. – DOI 10.18411/trnio-03-2023-308. – EDN QBTLBX.
269. Обобщение формулы Эйлера для непланарного графа / Н. К. Волосова, К. А. Волосов, А. К. Волосова [и др.]. – Москва : Российский Университет Транспорта, 2024. – 29 с. – EDN JZUSRБ.
270. Критерий, количество и структура кубических вычетов в поле \mathbb{Z}_p / Ю. Ф. Пастухов, Д. Ф. Пастухов, К. А. Волосов [и др.] // Вопросы обеспечения безопасности в киберпространстве : Материалы II Всероссийской научно-технической конференции, Махачкала, 10–11 мая 2024 года. – Махачкала: Типография ФОРМАТ, 2024. – С. 156-158. – EDN LCHKCJ.
271. Алгебраические методы шифрования : Учебное пособие к лекционным и практическим занятиям для студентов специальности 1-98 01 01 Компьютерная безопасность / Д. Ф. Пастухов, Н. К. Волосова, Ю. Ф. Пастухов [и др.]. – 3-е издание. – Москва : Учреждение образования «Полоцкий государственный университет, 2022. – 37 с. – EDN SNZVRK.
272. Моделирование систем. Лекции. Лабораторный практикум : Учебное пособие к лекционным и практическим занятиям для студентов специальности 1-40 01 01 Программное обеспечение информационных технологий / Д. Ф. Пастухов, Ю. Ф. Пастухов, Н. К. Волосова [и др.]. – 3-е дополненное. – Новополоцк : Учреждение образования «Полоцкий государственный университет имени Евфросинии Полоцкой», 2022. – 142 с. – EDN PYJINB.
273. Алгоритмы и программы для вычисления производных дробного порядка с двойной точностью: Учебное пособие к лекционным и практическим занятиям по предмету Специальные математические методы и функции для студентов специальности 1- 40 01 01 Программное обеспечение информационных технологий 1-39 03 02 Программируемые мобильные системы / Н. К. Волосова, К. А. Волосов, А. К. Волосова [и др.]. – 1-е издание. – Москва: Российский Университет Транспорта, 2024. – 57 с. – EDN TXNCSY.
274. Матричное решение краевой задачи для линейного обыкновенного дифференциального уравнения второго порядка с восьмым порядком погрешности /

- Н. К. Волосова, К. А. Волосов, А. К. Волосова [и др.]. – Москва: Российский университет транспорта (МИИТ), 2024. – 33 с. – EDN DDZPNH.
275. Сборник статей по гидродинамике / Н. К. Волосова, К. А. Волосов, А. К. Волосова [и др.]. – 2-е издание. – Москва: Московский государственный университет путей сообщения Императора Николая II, 2023. – 231 с. – EDN UDVEDI.
276. Несколько теорем о числах Кармайкла / Ю. Ф. Пастухов, Н. К. Волосова, К. А. Волосов [и др.]. – Москва : Российская открытая академия транспорта федерального государственного бюджетного образовательного учреждения высшего образования "Российский университет транспорта" (МИИТ), 2023. – 33 с. – EDN ONPGFT.
277. Метод последовательных функциональных компенсаций в задачах математической физики : Учебное пособие для практических занятий по курсу Уравнения математической физики / Н. К. Волосова, К. А. Волосов, А. К. Волосова [и др.]. – Москва : Полоцкий государственный университет, 2022. – 10 с. – EDN QQYOOA.
278. Этап конструирования математической модели аневризмы. Течения в каверне и противоречия в задаче в "закрытой" кювете / Н. К. Волосова, М. А. Басараб, А. К. Волосова [и др.] // Некоторые Актуальные проблемы современной математики математического образования: Материалы 74-й научной КОНФЕРЕНЦИИ «ГЕРЦЕНОВСКИЕ ЧТЕНИЯ 2021», Санкт-Петербург, 05–10 апреля 2021 года / Российская Академия Образования; Академия информатизации образования; Российский государственный педагогический университет им. А. И. Герцена, Кафедра математического анализа, Кафедра компьютерной инженерии и программной техники. – Санкт-Петербург: ООО "Издательство ВВМ", 2021. – С. 208–213. – EDN HREUQK.
279. Численные методы. Лекции. Численный практикум: учебное пособие / Д. Ф. Пастухов, Ю. Ф. Пастухов, Н. К. Волосова [и др.]. – 3-е издание, дополненное. – Новополоцк. 2021. – 237 с.
280. Пастухов, Д. Ф. Построение нестационарных моделей в оболочке ANSYS Fluent : учебное пособие / Д. Ф. Пастухов, Н. К. Волосова, Ю. Ф. Пастухов. – Новополоцк : Полоцкий государственный университет, 2018. – 45 с. – EDN SBWQRY.
281. Сборник статей по гидродинамике : Учебное пособие к лекционным и практическим занятиям для студентов специальности 1-40 01 01 Программное обеспечение информационных технологий / Н. К. Волосова, К. А. Волосов, А. К. Волосова [и др.]. – Москва : Полоцкий государственный университет, 2022. – 219 с. – EDN UAADIO.
282. О двух численных алгоритмах для решения конечномерной задачи Лагранжа на экстремум с ограничениями типа равенств : Учебное пособие для практических занятий по предметам Методы оптимизации и Математическое программирование / Н. К. Волосова, К. А. Волосов, А. К. Волосова [и др.]. – Москва : Полоцкий государственный университет, 2022. – 33 с. – EDN ZHJPU.

УДК 519. 6

ПРАКТИКУМ
ДЛЯ ЗАДАЧ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА

Волосов Константин Александрович, д.ф.-м.н., профессор
(МИИТ, г. Москва)

Волосова Наталья Константиновна
(МГТУ им. Н.Э. Баумана, г. Москва)

Волосова Александра Константиновна, к.ф.-м.н.
(МИИТ, г. Москва)

Пастухов Дмитрий Феликсович, к. ф.-м. н.

Пастухов Юрий Феликсович, к. ф.-м. н.

(Полоцкий государственный университет имени Евфросинии Полоцкой)

2026