

**МИНОБРНАУКИ РОССИИ**  
**Ярославский государственный университет им. П.Г. Демидова**

Кафедра нелинейной динамики

УТВЕРЖДАЮ

Декан математического факультета

\_\_\_\_\_ Нестеров П.Н.

20 мая 2025 г.

**Рабочая программа дисциплины**

**Суперкомпьютерные технологии и основы искусственного интеллекта**

Направление подготовки (специальности)  
01.03.02 Прикладная математика и информатика

Направленность (профиль)  
«Прикладное программирование и информационные технологии»

Форма обучения очная

Программа рассмотрена  
на заседании кафедры  
от 21.04.2025, протокол № 8

Программа одобрена НМК  
математического факультета  
протокол № 9 от 05.05.2025

## 1. Цели освоения дисциплины

Целями преподавания дисциплины «Суперкомпьютерные технологии и основы искусственного интеллекта» являются:

- знакомство слушателей с теоретическими основами и современными программно-аппаратными средствами для решения прикладных задач в следующих областях:

1. Классические высокопроизводительные вычисления (HPC) на примерах математического моделирования, комбинаторики и статистических методов
2. Машинное обучение с приложениями в области машинного зрения, обработки естественных языков, задач биоинформатики
3. Введение в искусственный интеллект и взаимодействие интеллектуальных агентов

## 2. Место дисциплины в структуре образовательной программы

Данная дисциплина относится к части образовательной программы, формируемой участниками образовательных отношений.

## 3. Планируемые результаты обучения по дисциплине, соотнесенные с планируемыми результатами освоения образовательной программы

Процесс изучения дисциплины направлен на формирование следующих компетенций в соответствии с ФГОС ВО, ООП ВО и приобретения следующих знаний, умений, навыков и (или) опыта деятельности:

Формируемая компетенция (код и формулировка)	Индикатор достижения компетенции (код и формулировка)	Перечень планируемых результатов обучения
<b>Профессиональные компетенции</b>		
<b>ПК-3</b> Способен к разработке и применению алгоритмических и программных решений в области системного и прикладного программного обеспечения	<b>И-ПК-3.1</b> Обладает устойчивыми знаниями в области разработки алгоритмов и программирования	<b>Знать:</b> - архитектуру кластерных вычислительных систем и архитектуру сопроцессоров для задач машинного обучения - методы решения задач математического моделирования и машинного обучения на современных вычислительных платформах
	<b>И-ПК-3.2</b> Имеет навыки разработки и реализации алгоритмов в области системного и прикладного программного обеспечения	<b>Владеть навыками:</b> - разработки программного обеспечения для задач математического моделирования и машинного обучения
	<b>И-ПК-3.3</b> Обладает способностью критического анализа и совершенствования разрабатываемых алгоритмов и программ	<b>Уметь:</b> - оценивать применимость вычислительных платформ для прикладных задач - выбирать эффективные алгоритмы и/или их библиотечные реализации для решения прикладных задач - тестировать полученный продукт, находить ошибки и исправлять их

#### 4. Объем, структура и содержание дисциплины

Общая трудоемкость дисциплины составляет **3** зачетные единицы, **108** акад. часов.

№ п/п	Темы (разделы) дисциплины, их содержание	Семестр	Виды учебных занятий, включая самостоятельную работу студентов, и их трудоемкость (в академических часах)						Формы текущего контроля успеваемости  Форма промежуточной аттестации (по семестрам)
			Контактная работа					самостоятельная работа	
			лекции	практические	лабораторные	консультации	аттестационные испытания		
1	Вводная лекция	7	4	6		1		4	Задания для самостоятельной работы
2	Высокопроизводительные вычисления	7	12	18		3		4	Практическое задание 1
3	Машинное обучение	7	8	12		3		4	Задания для самостоятельной работы
4	Искусственный интеллект	7	8	12		3		4	Практическое задание 2
							0.3	1.7	зачет
	ИТОГО		32	48		10	0.3	17.7	

#### Содержание разделов дисциплины:

##### Тема 1. Вводная лекция

1.1. Обзор курса и его мотивация. Работа с вычислительным кластером. Введение в Python notebooks / JupyterLab

1.2. Классические высокопроизводительные вычисления. Машинное обучение. Глубокое обучение. Суррогатные модели - синтез классического НРС и машинного обучения.

##### Тема 2. Высокопроизводительные вычисления

2.1. Современные архитектуры и программные модели (многоядерность, gpgpu, кластеры, - openmp, cuda, mpi), high performance vs high throughput/availability. OpenMP и параллелизм циклов

2.2. Параллелизм задач, виды и ограничения

2.3. Классические задачи НРС: дифференциальные уравнения - математическое моделирование. Решение уравнения теплопроводности.

2.4. Классические задачи НРС: комбинаторика, графы — рендеринг. Динамический параллелизм для деревьев

2.5. MPI и крупнозернистый параллелизм. CUDA и линейная алгебра. Быстрое преобразование Фурье

2.6 Классические задачи НРС: Монте-Карло - статистические методы. Параллельные генераторы случайных чисел

##### Тема 3. Машинное обучение

3.1. Формальная модель обучения, равномерная сходимость. Обзор фреймворков для машинного обучения / искусственного интеллекта.

3.2. Линейные предикторы. Дилемма смещения-сложности. VC-размерность. Фреймворки для ML/AI: Tensorflow, PyTorch

- 3.3. Бустинг. Метод опорных векторов. Кластеризация
- 3.4. Распознавание речи. Задачи геологии и сейсмологии.

#### **Тема 4. Искусственный интеллект**

- 4.1. Нейросети. Полносвязные и рекуррентные слои. Обработка естественных языков.
- 4.2. Сверточные нейросети. Задачи машинного зрения, детекция, трекинг.
- 4.3. Генеративно-состязательные сети.
- 4.4. Метрики качества детекторов и классификаторов.

### **5. Образовательные технологии, в том числе технологии электронного обучения и дистанционные образовательные технологии, используемые при осуществлении образовательного процесса по дисциплине**

В процессе обучения используются следующие образовательные технологии:

Вводная лекция – дает первое целостное представление о дисциплине и ориентирует студента в системе изучения данной дисциплины. Студенты знакомятся с назначением и задачами курса, его ролью и местом в системе учебных дисциплин и в системе подготовки в целом. Дается краткий обзор курса, история развития науки и практики, достижения в этой сфере, имена известных ученых, излагаются перспективные направления исследований. На этой лекции высказываются методические и организационные особенности работы в рамках данной дисциплины, а также дается анализ рекомендуемой учебно-методической литературы.

Академическая лекция с элементами лекции-беседы – последовательное изложение материала, осуществляемое преимущественно в виде монолога преподавателя. Элементы лекции-беседы обеспечивают контакт преподавателя с аудиторией, что позволяет привлекать внимание студентов к наиболее важным темам дисциплины, активно вовлекать их в учебный процесс, контролировать темп изложения учебного материала в зависимости от уровня его восприятия.

Практическое занятие – занятие, посвященное освоению конкретных умений и навыков по закреплению полученных на лекции знаний.

Консультации – вид учебных занятий, являющийся одной из форм контроля самостоятельной работы студентов. На консультациях по просьбе студентов рассматриваются наиболее сложные моменты при освоении материала дисциплины, преподаватель отвечает на вопросы студентов, которые возникают у них в процессе самостоятельной работы.

### **6. Перечень лицензионного и (или) свободно распространяемого программного обеспечения, используемого при осуществлении образовательного процесса по дисциплине**

В процессе осуществления образовательного процесса по дисциплине используются: для формирования материалов для текущего контроля успеваемости и проведения промежуточной аттестации, для формирования методических материалов по дисциплине:

- программы Microsoft Office;
- издательская система LaTeX;
- Adobe Acrobat Reader;
- Nvidia CUDA, Nvidia cuDNN;
- OpenMP, OpenMPI;
- Python, NumPy, SciPy, Pandas, OpenCV, PyTorch;
- YOLOv5.

## **7. Перечень современных профессиональных баз данных и информационных справочных систем, используемых при осуществлении образовательного процесса по дисциплине (при необходимости)**

В процессе осуществления образовательного процесса по дисциплине используются:

- Автоматизированная библиотечно-информационная система «БУКИ-NEXT» [http://www.lib.uniyl.ac.ru/opac/bk\\_cat\\_find.php](http://www.lib.uniyl.ac.ru/opac/bk_cat_find.php)
- Электронно-библиотечная система «Юрайт» <https://urait.ru>
- Электронно-библиотечная система «Лань» <http://e.lanbook.com/>
- Электронно-библиотечная система «Консультант Студента»: <https://www.studentlibrary.ru/>

## **8. Перечень основной и дополнительной учебной литературы, ресурсов информационно-телекоммуникационной сети «Интернет» (при необходимости), рекомендуемых для освоения дисциплины**

### **а) основная литература**

1. Боресков, А. В. Основы работы с технологией CUDA / Боресков А. В. , Харламов А. А. - Москва : ДМК Пресс, 2010. - 232 с. - ISBN 978-5-94074-578-5. - Текст : электронный // ЭБС "Консультант студента" : [сайт]. - URL : <https://www.studentlibrary.ru/book/ISBN9785940745785.html>
2. Доусон М. Програмируем на Python. / М. Доусон; [пер. с англ. В. Порицкого] - СПб.: Питер, 2016. - 414 с.

### **б) дополнительная литература**

1. Коршунов Д.А., Чернова Н.И. Сборник задач и упражнений по математической статистике. Учебное пособие. – Новосибирск: Нов ГУ, 2004.
2. Shai Shalev-Shwartz and Shai Ben-David Understanding Machine Learning: From Theory to Algorithms. Cambridge University press, 2014  
<https://www.cs.huji.ac.il/w~shais/UnderstandingMachineLearning/understanding-machine-learning-theory-algorithms.pdf>
3. И. В. Галыгина, Л. В. Галыгина Основы искусственного интеллекта. Лабораторный практикум — Санкт-Петербург: Лань, 2022. <https://reader.lanbook.com/book/261143>

### **в) ресурсы сети «Интернет» (при необходимости)**

1. MIT 6.S191 Introduction to Deep Learning <http://introtodeeplearning.com/>
2. PyTorch <https://pytorch.org/>
3. YOLOv5 <https://github.com/ultralytics/yolov5>

## **9. Материально-техническая база, необходимая для осуществления образовательного процесса по дисциплине**

Материально-техническая база, необходимая для осуществления образовательного процесса по дисциплине включает в свой состав специальные помещения:

- учебные аудитории для проведения занятий лекционного типа;
- компьютерные классы для проведения практических занятий (семинаров);
- учебные аудитории для проведения групповых и индивидуальных консультаций;
- учебные аудитории для проведения текущего контроля и промежуточной аттестации;
- помещения для самостоятельной работы;
- помещения для хранения и профилактического обслуживания технических средств обучения.

- вычислительный кластер ЯрГУ с программным обеспечением для параллельных вычислений и машинного обучения.

Специальные помещения укомплектованы средствами обучения, служащими для представления учебной информации большой аудитории.

Помещения для самостоятельной работы обучающихся оснащены компьютерной техникой с возможностью подключения к сети «Интернет» и обеспечением доступа к электронной информационно-образовательной среде ЯрГУ.

**Автор(ы):**

научный сотрудник, к.ф.-м.н,

Глызин Д.С.

**Приложение № 1 к рабочей программе дисциплины  
«Суперкомпьютерные технологии и основы искусственного интеллекта»**

**Фонд оценочных средств  
для проведения текущего контроля успеваемости  
и промежуточной аттестации студентов  
по дисциплине**

**1. Типовые контрольные задания и иные материалы,  
используемые в процессе текущего контроля успеваемости**

**Задания для самостоятельной работы (преподавателем не проверяются)**

**1. По теме «Введение»**

1. Найти оценку максимального правдоподобия параметра  $b$  равномерного распределения на отрезке  $[0, b]$
2. Найти оценку максимального правдоподобия параметра  $b > 0$  равномерного распределения на отрезке  $[-b, 0]$
3. Найти оценку максимального правдоподобия параметра  $b > 0$  равномерного распределения на отрезке  $[-b, b]$
4. Найти оценку максимального правдоподобия параметра  $b > 0$  равномерного распределения на отрезке  $[b, b+2]$
5. Найти оценку максимального правдоподобия параметра  $b > 0$  равномерного распределения на отрезке  $[b, 2b]$
6. Пусть  $X_1, \dots, X_n$  — выборка из равномерного распределения на отрезке  $[0, b]$ . С помощью статистики  $X_{(n)}$  постройте точный доверительный интервал уровня  $1-\epsilon$  для параметра  $b$ .

**2. По теме «Машинное обучение»**

Рассмотрим классификаторы, возвращающие 0 для всех точек внутри прямоугольной области со сторонами, параллельными осям координат. Предположим также, что выполнено условие реализуемости.

1. Покажите, что алгоритм, возвращающий минимальный прямоугольник, содержащий позитивные метки из обучающей выборки, минимизирует эмпирический риск.
2. Покажите, что если размер обучающей выборки больше или равен  $4 \log(4/\delta) / \epsilon$  то с вероятностью по меньшей мере  $1-\delta$  этот алгоритм возвращает гипотезу с ошибкой не более  $\epsilon$
3. Распространите результат из п.2 на пространство  $R^d$ .
4. Покажите, что трудоемкость данного алгоритма полиномиальна по размерности  $d$ , по  $1/\epsilon$ , а также по  $\log(1/\delta)$ .

**Практическое задание 1 (по теме «Высокопроизводительные вычисления»)**

Вариант 1. Решите численно с применением технологии OpenMP уравнение теплопроводности в квадратной области с заданными граничными условиями.

Вариант 2. Решите численно с применением технологии MPI уравнение теплопроводности в квадратной области с заданными граничными условиями.

Вариант 3. Решите численно с применением технологии CUDA уравнение теплопроводности в квадратной области с заданными граничными условиями.

**Практическое задание 2 (по теме «Искусственный интеллект»)**

Выбрав один из открытых наборов данных для тренировки машинного зрения (<https://public.roboflow.com/>) произведите тренировку в сервисе Google Colab (<https://colab.research.google.com>) нейросети YOLOv5s (<https://github.com/ultralytics/yolov5>)

Проанализируйте полученные результаты с точки зрения качества детекции и классификации.

## **2. Список вопросов и (или) заданий для проведения промежуточной аттестации**

Список вопросов к зачету (проведение зачета подразумевает наличие у студента разработанных программ для практических заданий 1 и 2):

по практическому заданию 1:

1. Объясните выбор цикла для распараллеливания в решении задачи математического моделирования

2. Измените направление распараллеливания (по строкам или по столбцам) расчета матрицы состояния

3. Измените расположение граничного условия в программе

4. Проинтерпретируйте результаты ускорения работы параллельной программы по практическому заданию 2:

5. Проинтерпретируйте кривую precision в результатах обучения нейросети

6. Проинтерпретируйте кривую recall в результатах обучения нейросети

7. Проинтерпретируйте кривую mAP в результатах обучения нейросети

8. Объясните различия кривых mAP для обучающей и валидационной выборок.



## Приложение № 2 к рабочей программе дисциплины «Суперкомпьютерные технологии и основы искусственного интеллекта»

### Методические указания для студентов по освоению дисциплины

Курс состоит из лекций и практических занятий. Лекции обеспечивают теоретическую подготовку студентов, создают представление о наборе решаемых современными вычислительными средствами задач, о том, какие результаты могут быть сформулированы на вероятностном языке строго. Практические занятия требуют от студентов базовых навыков программирования и проходят путем демонстрации решения типовых задач средствами языков C++ и Python с использованием соответствующих библиотек. Крайне важной является самостоятельная работа над программным кодом и наработка навыков создания воспроизводимых расчетов с анализом полученных характеристик.

Методические указания по разработке программы для параллельного решения задачи теплопроводности (практическое задание 1):

### Решение двумерного уравнения теплопроводности с использованием OpenMP

Глызин Дмитрий Сергеевич  
[glyzin@gmail.com](mailto:glyzin@gmail.com)  
<http://parallel.ncycle.org>

ЯрГУ им. П.Г. Демидова

I семестр 2011/2012 учебного года



## Метод

Теперь введем обозначения для узлов сетки:  $x_i = i\Delta x$ ,  $y_j = j\Delta y$ ,  $t_k = k\Delta t$ . Значение функции в узле обозначим  $u_{i,j,k} = u(x_i, y_j, t_k)$ . Как известно, при выполнении условия

$$\left( \frac{1}{\Delta x^2} + \frac{1}{\Delta y^2} \right) \Delta t \leq \frac{1}{2} \quad (7)$$

решение конечно-разностной аппроксимации

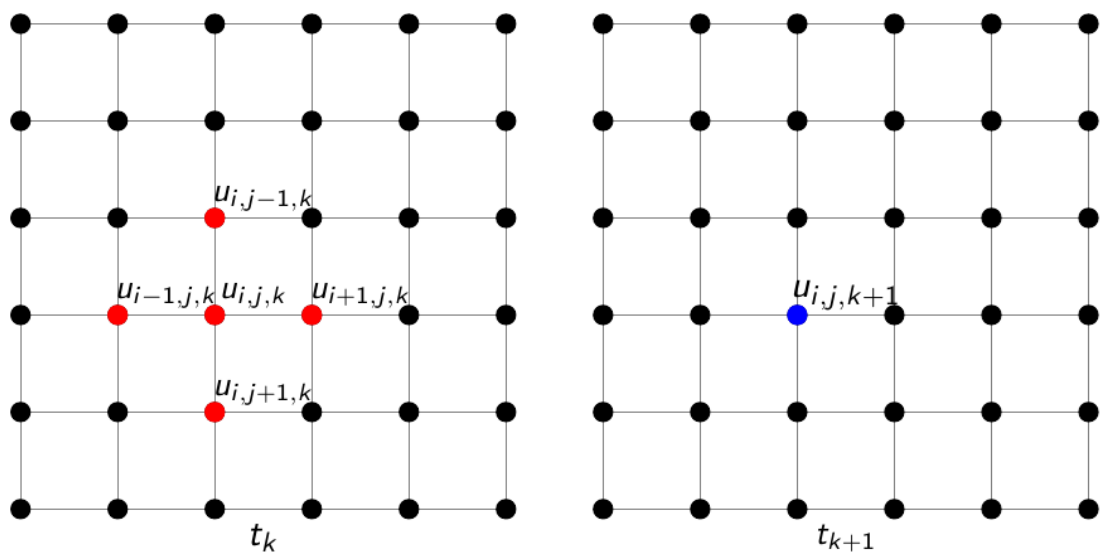
$$\frac{u_{i,j,k+1} - u_{i,j,k}}{\Delta t} = \frac{u_{i+1,j,k} + u_{i-1,j,k} - 2u_{i,j,k}}{\Delta x^2} + \frac{u_{i,j+1,k} + u_{i,j-1,k} - 2u_{i,j,k}}{\Delta y^2} \quad (8)$$

сходится с измельчением шагов к решению задачи (1)-(3). Из уравнения (8) можно выразить  $u_{i,j,k+1}$  через значения в предыдущий момент времени.

Navigation icons: back, forward, search, etc.

## Метод

Для того, чтобы найти значение  $u_{i,j,k+1}$  по формуле (8) в точке, отмеченной синим, потребуется пять значений в точках, отмеченных красным:



Navigation icons: back, forward, search, etc.

## Параллелизм

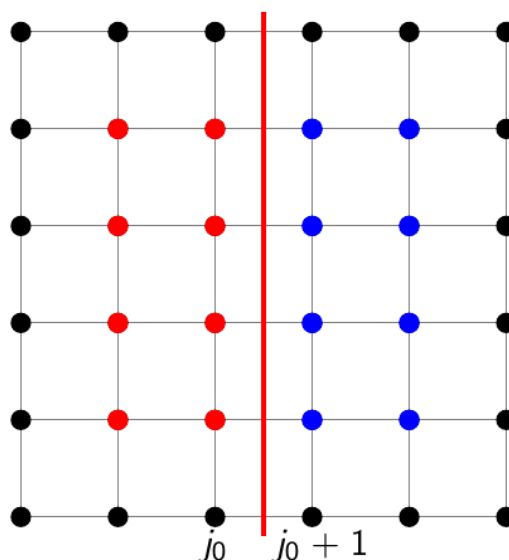
Для параллельного решения задачи воспользуемся декомпозицией области. Пусть у нас имеется  $p$  процессоров, разделим пространственную сетку по одной из координат на  $p$  прямоугольных блоков одинаковой величины. На каждой итерации по времени вычисление значений элементов блока не зависит от значений в других блоках, за исключением пограничных рядов.

Таким образом, итерации каждого блока можно выполнять на отдельном процессоре, обмениваясь в конце итерации с "соседями" значениями из пограничных рядов.

◀ ◻ ▶ ◀ ◻ ▶ ◀ ≡ ▶ ◀ ≡ ▶ ≡ ≡ ≡ ≡ ↺ ↻

## Параллелизм

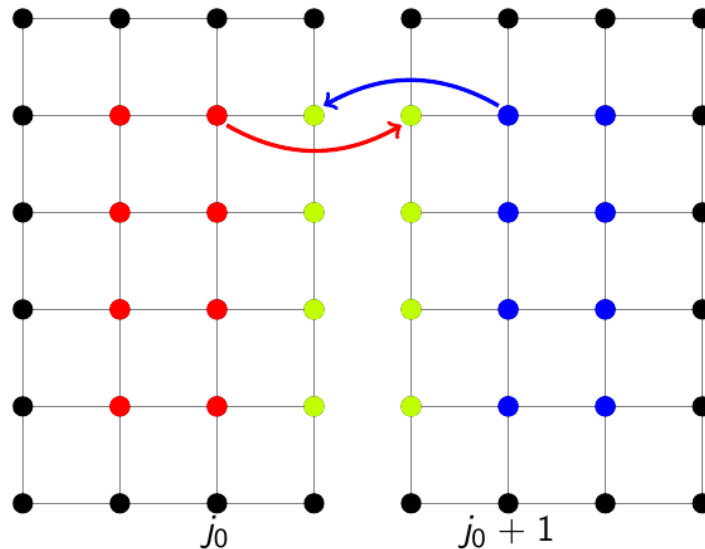
Разделим область по вертикали между столбцами  $j_0$  и  $j_0 + 1$ . Черными обозначены граничные точки, не требующие пересчета, красными и синими — точки, обрабатываемые первым и вторым процессорами.



◀ ◻ ▶ ◀ ◻ ▶ ◀ ≡ ▶ ◀ ≡ ▶ ≡ ≡ ≡ ≡ ↺ ↻

## Параллелизм

Дополним матрицы каждого из процессоров фиктивными столбцами с нужной стороны, и после каждой итерации будем копировать в них крайний вычисленный столбец соседнего процессора:



## Требования к программе

**Программа** должна компилироваться либо в Visual Studio 2008 (необходимо прислать проект целиком), либо с помощью TDM-GCC (или gcc в Linux), в этих случаях нужно прислать make-файл.

**Ввод параметров** осуществляется через файл "input.txt", лежащий в той же директории, что исполняемый файл. Каждая строка файла содержит по одному числу в следующем порядке:

- ①  $p$ , количество нитей, на которое распараллеливается вычисление ( $1 \leq p \leq 4$ )
- ②  $N$ , количество точек сетки по каждому направлению
- ③  $\Delta t$ , шаг по времени
- ④  $T$ , длина промежутка времени, на котором решается задача
- ⑤ filename, имя файла для вывода данных

## Требования к программе

**Параметры** должны удовлетворять условию  $4(N - 1)^2 \Delta t \leq 1$ , в противном случае программа должна выводить сообщение об ошибке и останавливаться.

**Время вычислений** нужно измерить и вывести в файл filename первой строкой, в начале которой нужно поставить знак # (знак комментария в gnuplot). Это время не должно учитывать ввод или вывод.



## Требования к программе

**Выводить приближение**  $u(T, x, y)$  в файл filename нужно в соответствии с требованиями gnuplot для хранения трехмерных графиков, т.е. в виде:

$x_0 \quad y_0 \quad u(T, x_0, y_0)$

$x_0 \quad y_1 \quad u(T, x_0, y_1)$

...

$x_0 \quad y_{N-1} \quad u(T, x_0, y_{N-1})$

пустая строка

$x_1 \quad y_0 \quad u(T, x_1, y_0)$

$x_1 \quad y_1 \quad u(T, x_1, y_1)$

...

$x_1 \quad y_{N-1} \quad u(T, x_1, y_{N-1})$

пустая строка

...

$x_{N-1} \quad y_0 \quad u(T, x_{N-1}, y_0)$

$x_{N-1} \quad y_1 \quad u(T, x_{N-1}, y_1)$

...

$x_{N-1} \quad y_{N-1} \quad u(T, x_{N-1}, y_{N-1})$



## Полезные советы

- Сравнивайте результаты, которые выводит программа при вычислении в одну нить с вычислениями в несколько нитей. Результаты должны быть идентичными, это простой способ проверить корректность работы параллельной программы.
- Поскольку параллелизм задачи при большом размере достаточно высок, ускорение на двухядерном процессоре при запуске в одну и в две нити должно быть близко к 1.5. Это хороший признак того, что программа работает эффективно.
- При работе с массивами избегайте лишнего копирования больших объемов данных. В случае, когда требуется поменять массивы ролями, можно обойтись обменом указателей.
- Во избежание лишних проверок можно считать, что  $N - 2$  делится на  $p$  нацело.