Ознакомительная учебная практика

Макаров П. А., доцент кафедры ИСиТ,
Сыктывкарский лесной институт (филиал)
ФГБОУ ВО «Санкт-Петербургский государственный
лесотехнический университет имени С. М. Кирова»

Сыктывкар, 2026

Аннотация

Методические материалы по ознакомительной учебной практике. Предназначены для студентов, обучающихся по направлению бакалавриата 09.03.02 (профили «Информационные системы и технологии» и «Информационные системы и технологии в бизнесе») на транспортно-технологическом факультете Сыктывкарского лесного института. Подготовлено для групп 325 и 825, проходящих практику в третьем семестре в 2025-2026 учебном году. Материалы разработаны на основе методических указаний [1].

Содержание

1 Общие положения **2**
1.1 Цели и задачи . . . 2
1.2 Связь с другими дисциплинами . . . 2
1.3 План работы . . . 3
1.4 Аттестация . . . 3
1.5 Структура и содержание отчёта . . . 3

2 Теоретический минимум **4**

3 Научные поисковые системы **4**

4 Программные инструменты **5**
4.1 TEX и LATEX . . . 5
4.2 Работа с библиографией . . . 5
4.3 Научная графика . . . 6
4.4 Языки программирования, библиотеки, трансляторы, фреймворки и отладчики . . . 6

5 Задания на практику **7**
5.1 Темы рефератов для теоретической части 7
5.2 Практические задания 8
5.2.1 Типовые алгоритмы обработки одномерных массивов данных (Р1) 9
5.2.2 Операции с матрицами (Р2) 10
5.2.3 Работа с файлами данных (Р3) 11
5.2.4 Элементы компьютерной графики (Р4) 12
5.2.5 Расчёт динамики материальной точки (Р5) 16
5.2.6 Пример выполнения задания Р5 24
5.3 Варианты заданий на практику 32

Список использованных источников **35**

1 Общие положения

1.1 Цели и задачи

Цель практики — закрепление и углубление знаний, полученных студентами в процессе теоретического обучения; приобретение и формирование необходимых навыков и опыта практической работы по избранной специальности; усвоение приёмов, методов и способов обработки, представления и интерпретации результатов проведенных практических исследований; развитие творческих способностей.

Задачи практики:

1. овладение методами работы в различных операционных системах;
2. совершенствование навыков использования офисных программ.
3. освоение принципов построения файловых систем;
4. овладение основами программирования и использования объектно-ориентированных систем визуального программирования.

1.2 Связь с другими дисциплинами

Учебная практика готовит студентов к более глубокому усвоению теоретических знаний и закреплению их на практике. Предполагается знакомство студентов со следующими дисциплинами: «Введение в профессиональную деятельность», «Информатика».

Ознакомительная практика предназначена для закрепления знаний о методах и технологиях программирования, а также ознакомления студентов с мето-

дами работы системного и прикладного программного обеспечения. Практика даёт базовые навыки работы на ЭВМ, программирования и алгоритмизации, а также проведения работ по проектирования информационных систем.

1.3 План работы

1. Введение (2 ч.);
2. Изучение необходимой теории (4 ч.);
3. Непосредственное выполнение практических заданий (4 ч.);
4. Подготовка отчёта (2 ч.);
5. Защита отчёта.

Общий объём практики: 12 ч. в форме практических занятий.

1.4 Аттестация

Согласно учебному плану по практике предусмотрен дифференцированный зачёт. Основное требование для получения зачёта — **защита выполненного отчёта о проделанной работе.**

1.5 Структура и содержание отчёта

Структура отчёта:

1. Титульный лист;
2. Введение (1-2 стр.);
3. Теоретическая часть ($\geqslant$ 15 стр.);
4. Практическая часть;
5. Заключение;
6. Список использованных источников и литературы (не менее 10 ед.).

Во введении ставятся цели и определяются задачи при прохождении практики.

Теоретическая часть представляет собой реферат на заданную тему объемом не менее 15 страниц. В данном разделе раскрывается понятие и основные характеристики информационной технологии или явления, описывается их роль, структура, история развития, классы и виды.

Практическая часть представляет собой описание программных реализаций пяти индивидуальных заданий по следующим темам:

1. Типовые алгоритмы обработки одномерных массивов данных.
2. Операции с матрицами.
3. Работа с файлами данных.
4. Элементы компьютерной графики.
5. Расчёт динамики материальной точки.

Оформление каждого задания в отчёте должно включать следующие элементы:

1. Точная формулировка постановки задачи;
2. Системный анализ проблемы и аналитическое решение там где это необходимо;
3. Блок-схема и словесное описание реализуемого алгоритма;
4. Код программы на языках Pascal, C, C++, Python (первые три — на выбор, последний — по желанию дополнительно к выбранному);
5. Распечатки или скриншоты сеансов работы программы.

В заключении отражаются результаты прохождения практики и подводятся итоги.

2 Теоретический минимум

В качестве основной литературы рекомендуются источники [2—7]. Дополнительно могут использоваться книги [8—10] и любые другие доступные источники.

Весьма полезным является ресурс [11], представляющий собой удобно организованную дорожную карту (roadmap) изучения всей отрасли Computer Science в целом.

3 Научные поисковые системы

- Google Академия scholar.google.com [12];
- Mendeley www.mendeley.com [13];
- arXiv arxiv.org [14];
- Научная электронная библиотека www.elibrary.ru [15];
- Web of Science webofknowledge.com [16];
- Scopus www.scopus.com [17];
- ScienceDirect www.sciencedirect.com [18];

- Springer Link link.springer.com [19];
- Электронная библиотека Института инженеров по электротехнике и электронике IEEE Xplore [20];
- Интеллектуальная Система Тематического Исследования НАукометрических данных ИСТИНА [21];
- MathSciNet mathscinet.ams.org [22];
- Проект Евклид projecteuclid.org [23];
- MathNet www.mathnet.ru [24];

4 Программные инструменты

4.1 TEX и LATEX

Лично я изучал LATEX по книге Котельникова и Чеботарева [25], поэтому рекомендую прочесть из неё по крайней мере главы 1–6. Ещё одна классическая книга по набору и вёрстке текстов в системе LATEX написана Львовским [26]. По большому счёту, этих источников вполне достаточно, однако, возможно, вам больше понравится (или будет более полезной) книга Столярова [27] (по крайней мере, она гораздо короче). Кроме того, настойчиво рекомендую ознакомиться с книгой Дональда Кнута — создателя системы TEX [28].

Также перед началом работы очень советую прочесть Краткий курс благородного набора [29]. Ещё один крайне полезный при наборе документов ресурс — Detexify [30], созданный Дэниелем Киршем.

Файлы с расширением `.tex` — это обычные текстовые файлы (написанные на языке разметки текста LATEX). Открывать и редактировать их можно с помощью любого текстового редактора[1], но я советую Sublime Text [31].

Для работы с издательской системой LATEX на локальном компьютере необходимо установить соответствующий набор программного обеспечения. Рекомендуется использовать актуальный дистрибутив TeX Live [32]. В настоящее время существуют и различные онлайн ресурсы для работы с системой LATEX, в частности проект Overleaf [33].

4.2 Работа с библиографией

Для работы с библиографией в LATEX есть две возможности:

[1] Ни в коем случае не пытайтесь использовать для этого MS Word или что-то подобное!

1. ручной набор списка литературы текущего документа с помощью окружения `thebibliography`;
2. создание библиографической базы данных и автоматическая генерация списка литературы.

Во втором случае практически стандартом является связка двух пакетов: BibLaTeX [34] и Biber [35] (вариант использования BibTeX [36] ещё встречается, но не является рекомендуемым). Вместе с тем, вручную составлять достаточно объёмные библиографические базы данных бывает весьма затруднительно, поэтому используются системы управления библиографической информацией, такие как JabRef [37] и Mendeley Desktop [38].

4.3 Научная графика

Одним из самых широко используемых инструментов для визуализации данных является Gnuplot [39]. Официальная документация проекта доступна по данной ссылке [40]. Также существует страница с демонстрационными примерами [41]. Кроме этих, имеется огромное множество и других источников, одно перечисление которых — весьма трудоёмкая задача. Поэтому ограничимся только ещё одной полезной ссылкой [42].

Ещё один мощный инструмент создания научной графики высокого качества — язык описания векторной гафики Asymptote [43]. Официальная документация доступна по данной ссылке [44]. Крайне рекомендуются для изучения монографии Ю. М. Волченко [45] и P. Ivaldi [46]. В сети есть по-крайней мере две обширных галерии примеров: официальная (англоязычная) [47] и неофициальная (франкоязычная) [48]. В настоящее время существует даже вебприложение [49], позволяющее запускать программы на языке Asymptote непосредственно в браузере.

4.4 Языки программирования, библиотеки, трансляторы, фреймворки и отладчики

- C [7].
- C++ [50, 51].
- GSL [52].
- GCC [53].
- ROOT [54].

- GDB [55].

Для выполнения практических заданий подготовлен сервер под управлением UNIX-подобной ОС Ubuntu Linux. Доступ к серверу возможен с помощью любого браузера по адресу https://mpa.komisc.ru. Для подключения к серверу необходимо ввести имя своего пользователя и пароль, примеры которых можно найти в таблице 1. Также доступ к серверу возможен по протоколу `SSH` (ip address: 62.76.156.201, port: 2277).

Замечание 1. *При вводе пароля нет причин беспокоиться о том, что он не отображается при наборе. Это совершенно естественная ситуация, связанная с требованиями безопасности. Постарайтесь набрать логин и пароль без ошибок, после чего вы без каких-либо проблем войдёте на сервер.*

Таблица 1: Примеры логинов и паролей

Фамилия И. О.	`login`	`password`
Иванов И. И.	`ivanovii`	`iiivanov`
Петров П. П.	`petrovpp`	`pppetrov`
Сидоров С. С.	`sidorovss`	`sssidorov`

При отсутствии опыта работы с UNIX-подобными системами следует ознакомиться с параграфами 1, 2 и 4 пособия Столярова [56] или раздел 1.2 из книги [2] того же автора.

5 Задания на практику

5.1 Темы рефератов для теоретической части

1. Информационные технологии как явление (этапы развития, классификации, состав, структура и свойства).
2. Интеллектуальные и экспертные системы. Базы знаний.
3. САПР (понятие, принципы, классификации, обзор по отраслям).
4. Компьютерная графика.
5. Компьютерные сети и Интернет. Облачные сервисы.
6. Корпоративные информационные системы.
7. Основы веб-разработки: фронтенд и бэкенд.
8. История развития и поколения ЭВМ.

9. Архитектура современных компьютеров и их программное обеспечение.
10. Архитектуры операционных систем.
11. Контейнеризация и виртуализация: от виртуальных машин к Docker и оркестрации.
12. Основы защиты информации.
13. Компьютерные вирусы.
14. Введение в кибербезопасность: основы криптографии, уязвимости кода (OWASP Top 10) и этичный хакинг.
15. FLOSS — свободное программное обеспечение с открытым исходным кодом.
16. Алгоритмы.
17. Развитие языков программирования.
18. Прикладное и системное программирование.
19. Основные элементы языков программирования.
20. Основы параллельного программирования.
21. Простые и составные типы данных.
22. Статическая и динамическая память. Массивы элементов.
23. Процедуры и функции.
24. Стандарты языков C и C++.
25. Схема работы, организация и варианты трансляторов языков C и C++.
26. Обзор возможностей библиотеки GSL для вычислительных задач.
27. Системы контроля версий: от основ ветвления до стратегий ведения проектов.
28. DevOps и CI/CD: автоматизация сборки, тестирования и развёртывания программ.
29. Мобильная разработка: нативные (Kotlin/Swift) и кроссплатформенные (Flutter/React Native) подходы.

5.2 Практические задания

Выполнение практические заданий состоит в написании исходных текстов программ на языках программирования Pascal или C либо C++ и демонстрации преподавателю работы готовых приложений на нескольких примерах. При необходимости следует пояснить алгоритм работы программы, руководствуясь её исходным кодом. Каждое задание представляет собой некую проблему, исследование которой предполагает выполнение следующих этапов:

1. изучение необходимой литературы;
2. выяснение подробностей условия задания;
3. системный анализ и аналитическое решение задачи;
4. разработка программного алгоритма;
5. написание и отладка программ;
6. тестирование, проверка результатов на корректность и выявление ограничений разработанной реализации алгоритма;
7. подготовка дополнительных скриптов, позволяющих получить изображения и/или анимацию, там где это требуется;
8. написание отчёта о результатах проделанной работы.

Для решения представленных задач не принципиальны архитектура вычислительной машины, конкретная операционная система, тип и версия компилятора и/или IDE. Вы можете выбрать любое наиболее удобное для вас решение, но обязательно учитывайте возможность продемонстрировать результаты своей работы в компьютерных лабораториях СЛИ. В случае затруднений в выборе, рекомендуется пользоваться сервером, описанным в разделе 4.4.

При написании исходных текстов помните, что они должны быть грамотными и, по возможности, простыми. Имена переменным и функциям следует выбирать осмысленными, исходя из их функционального назначения. Не пренебрегайте возможностью использовать лаконичные и ёмкие комментарии, которые помогут лучше разобраться в коде вашей программы. Старайтесь красиво форматировать исходный код, это значительно упростит его восприятие и лично вами, и другими людьми. Продумывайте интерфейс ваших программ так, чтобы неподготовленный пользователь смог понять их предназначение и научиться ими правильно пользоваться. Кроме того, необходимо предусмотреть “внештатные” ситуации, которые могут возникнуть при исполнении ваших программ, и включить в исходный код их обработку.

5.2.1 Типовые алгоритмы обработки одномерных массивов данных (P1)

Программа работает с одномерным массивом данных (см. Таблицу 2) размерности N, сформированным одним из способов, перечисленных в Таблице 3.

Независимо от способа формирования массива, программа должна контролировать корректность данных, над которыми выполняется работа.

Требуется выполнить следующие задачи:

Таблица 2: Типы данных для заданий P1 (п. 5.2.1) и P2 (п. 5.2.2)

Вариант	Тип	Вариант	Тип
T1	Целые $\geqslant 0$	T4	Вещественные $\geqslant 0$
T2	Целые > 0	T5	Вещественные > 0
T3	Целые	T6	Вещественные

Таблица 3: Способы формирования входных данных для задания P1 (п. 5.2.1)

Вариант	Метод
I1	Ручной ввод данных в процессе выполнения программы
I2	Передача данных в качестве аргументов командной строки
I3	Генерация случайных значений

1. Определить максимальный и минимальный элементы массива, а также все отвечающие им индексы;
2. Вычислить сумму и прозведение всех элементов массива;
3. Найти среднее арифметическое и среднее геометрическое значения;
4. Для типов T1 – T3 отыскать количество чётных и нечётных элементов. Для типов T4 – T6 выполнить тоже самое для значений элементов массива, округлённых в сторону ближайшего целого;
5. Реализовать поиск в массиве значения, заданного пользователем программы. Вывести максимально подробную информацию о данном элементе: количество вхождений и соответствующие индексы;
6. Отсортировать массив в порядке возрастания и убывания значений элементов одним из методов, перечисленных в Таблице 4.

Таблица 4: Методы сортировки массивов [57] для задания P1 (п. 5.2.1)

Вариант	Метод сортировки	Вариант	Метод сортировки
M1	Вставками (Insertion)	M5	Быстрая (Quick)
M2	Выбором (Selection)	M6	Пирамидальная (Heap)
M3	Пузырьком (Bubble)	M7	Поразрядная (Radix)
M4	Подсчётом (Counting)	M8	Слиянием (Merge)

5.2.2 Операции с матрицами (P2)

Программа работает с двумя числовыми матрицами (допустимые типы данных перечислены в Таблице 2) произвольной размерности $M \times N$ (где M — это

число строк, а N — число столбцов матрицы). Все необходимые для работы данные должны вводиться пользователем во время исполнения программы.

Необходимо решить следующие задачи:

1. Определить ранги (rank) исходных матриц.
2. Выполнить их транспонирование.
3. Проверить, являются ли данные матрицы квадратными. В случае квадратных матриц найти их определители (determinant) и следы (trace), а также если возможно вычислить обратные матрицы.
4. Установить возможность в отношении введённых матриц следующих операций: проверка на равенство, сложение, произведение. Выполнить все допустимые из них.

5.2.3 Работа с файлами данных (P3)

Перед выполнением данных заданий рекомендуется ознакомиться с четвёртой главой пособия Богатырёва А.С. [58], доступного по данной ссылке.

Сами задания представляют собой следующие задачи, заимствованные из [58]. Номер конкретной задачи, соответствующей тому или иному варианту см. в Таблице 7.

1. Напишите программу подсчёта строк и символов в файле. Указание: надо подсчитать число символов '`\n`' в файле и учесть, что последняя строка файла может не иметь этого символа на конце. Поэтому если последний символ файла (тот, который вы прочитаете самым последним) не есть '`\n`', то добавьте к счетчику строк 1.
2. Напишите программу подсчёта количества вхождений каждого из символов алфавита в файл и печатающую результат в виде таблицы в 4 колонки. (Указание: заведите массив из 256 счётчиков. Для больших файлов счетчики должны быть типа `long`).
3. Напишите программу, которая распечатывает самую длинную строку из файла ввода и её длину.
4. Напишите программу, которая выдает `n`-ую строку файла. Номер строки и имя файла задаются как аргументы `main()`.
5. Составьте программу, которая распечатывает последние `n` строк файла ввода.
6. Напишите программу, которая делит входной файл на файлы по `n` строк в каждом.

7. Напишите программу, которая читает 2 файла и печатает их вперемежку: одна строка из первого файла, другая — из второго. Придумайте, как поступить, если файлы содержат разное число строк.
8. Напишите программу сравнения двух файлов, которая будет печатать первую из различающихся строк и позицию символа, в котором они различаются.
9. Напишите программу удаления `n`-ой строки из файла и её вставки после `m`-ой строки.
10. Напишите программу, которая будет выдавать подряд в стандартный вывод все файлы, чьи имена указаны в аргументах командной строки. Используйте `argc` для организации цикла. Добавьте сквозную нумерацию строк и печать номера строки.

5.2.4 Элементы компьютерной графики (P4)

Необходимо написать программу, выполняющую расчёт функции вида (конкретный вид функций для различных вариантов перечислен в Таблице 5)

$$y = y(x; z), \tag{1}$$

где x — независимая переменная, а z — некий набор параметров, при которых требуется произвести вычисления.

Программа должна запускаться в консоли с тремя аргументами следующим образом:

```
$ ./calc a b N
```

Здесь `a` и `b` — это границы изменения независимой переменной $x \in [a, b]$, а `N` — это количество вариантов различных наборов z.

После запуска, программа должна запросить пользователя ввести все возможные значения z, выполнить для них требуемые расчёты и сохранить их результаты в текстовые файлы с именами `1.txt`, ..., `N.txt` в формате таблиц вида:

```
#z = ...
a    y(a;z)
a+h  y(a+h;z)
.
```

.

`b  y(b;z)`

Шаг изменения h независимой переменной x всегда принимать равным

$$h = \frac{b-a}{10^3}. \tag{2}$$

Все вычисления выполнять с относительной погрешностью

$$\delta y = \frac{\Delta y}{y} \leqslant 10^{-3}. \tag{3}$$

После выполнения всех расчётов программа должна вывести на экран два значения `m` и `M`, которые определяются как минимальное и максимальное значения y, полученные в ходе всех вычислений

$$m = \min_{i=1,N} y(x; z_i), \quad M = \max_{i=1,N} y(x; z_i), \quad \forall x \in [a, b]. \tag{4}$$

Полученные в результате расчёта файлы `1.txt`, ..., `N.txt`, а также значения `m` и `M` использовать для построения графиков $y(x; z)$ с помощью программ Gnuplot или Asymptote, описанных в разделе 5.2.4, либо других доступных инструментов.

Таблица 5: Функции для задания P4 (п. 5.2.4)

Вариант	Функция	Вариант	Функция
F01	$T(\varphi_0;\ell) = 4\sqrt{\frac{\ell}{g}}\mathcal{K}\left(\sin\frac{\varphi_0}{2}\right)$	F06	$U(r;\varepsilon,\sigma) = 4\,\varepsilon\left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6}\right]$
F02	$f(E;\mu,T) = \frac{1}{\exp\left(\frac{E-\mu}{kT}\right)+1}$	F07	$U(r;A,B,C) = Ae^{-Br} - \frac{C}{r^6}$
F03	$f(E;\mu,T) = \frac{1}{\exp\left(\frac{E-\mu}{kT}\right)-1}$	F08	$U(r;D_e,k_e,r_e) = D_e\left[1-e^{-a(r-r_e)}\right]^2$
F04	$U(r;A) = -\frac{U_0}{1+\exp\left(\frac{r-R}{a}\right)}$	F09	$f(v;m,T) = \sqrt{\frac{2}{\pi}}\frac{v^2}{v_T^3}\exp\left(-\frac{v^2}{2v_T^2}\right)$
F05	$M(H;T,\mu_0,n) = \mu_0 n\,\mathcal{L}\left(\frac{\mu_0 H}{kT}\right)$	F10	$\phi(r;q,\lambda_D) = \frac{q}{4\pi\varepsilon_0}\frac{e^{-r/\lambda_D}}{r}$

F01. Описывает период колебаний T математического маятника, совершающего колебания в однородном поле силы тяжести на невесомой нерастяжимой нити длины ℓ. Здесь $g = 9.8$ м/с^2 — ускорение свободного падения, φ_0 — начальный угол отклонения маятника от вертикали, а $\mathcal{K}(x)$ — полный эллиптический интеграл первого рода.

Известно, что $\mathcal{K}(x)$ можно представить в виде ряда

$$\mathcal{K}(x) = \frac{\pi}{2}\left(1 + \frac{1^2}{2^2}\,x^2 + \frac{1^2\cdot 3^2}{2^2\cdot 4^2}\,x^4 + \frac{1^2\cdot 3^2\cdot 5^2}{2^2\cdot 4^2\cdot 6^2}\,x^6 + \ldots\right), \quad x^2 < 1. \qquad (5)$$

F02. Функция распределения Ферми — Дирака. Представляет собой вероятность заполнения энергетических уровней E тождественными частицами-фермионами (частицы с полуцелым спином, подчиняющиеся принципу запрета Паули), находящимися в термодинамическом равновесии. Здесь T — абсолютная (термодинамическая) температура системы, μ — химический потенциал (равный энергии Ферми E_F системы при нуле температуры), а $k = 1.38 \cdot 10^{-23}$ Дж/К — постоянная Больцмана.

F03. Функция распределения Бозе — Эйнштейна (функция распределения идеального квантового газа, подчиняющегося статистике Бозе — Эйнштейна). Даёт вероятность заполнения тождественными частицами-бозонами (частицы с нулевым или целочисленным спином) энергетических уровней E при условии, что взаимодействие частиц в системе слабое и им можно пренебречь. Смысл всех величин такой же как для функции F02. Химический потенциал для Бозе-газа с постоянным числом частиц принимает отрицательные и большие по модулю значения. Химический потенциал системы бозонов с переменным числом частиц равен нулю.

F04. Потенциал Вудса — Саксона. Часто применяется в оболочечных моделях ядра для расчёта центрально-симметричной части потенциальной энергии нуклона в поле ядерных сил. Здесь $U_0 \sim 50$ МэВ — глубина потенциальной ямы; $a \sim 0.5$ фм — параметр диффузности, характеризующий размытие края потенциальной ямы; $R = r_0 A^{1/3}$ — радиус ядра, где $r_0 \approx 1.3$ фм — среднее расстояние между нуклонами в ядре, а A — массовое число ядра.

F05. Намагниченность M идеального газа магнитных диполей с магнитными моментами μ_0 и концентрацией n, находящихся в магнитном поле с напряжён-

ностью H при температуре T согласно классической теории парамагнетизма. Здесь $k = 1.38 \cdot 10^{-23}$ Дж/К — постоянная Больцмана, а $\mathcal{L}(x)$ — функция Ланжевена.

Известно, что $\mathcal{L}(x)$ можно представить в виде

$$\mathcal{L}(x) = \frac{e^x + e^{-x}}{e^x - e^{-x}} - \frac{1}{x}. \tag{6}$$

F06. Потенциал Леннарда — Джонса. Простая модель парного взаимодействия неполярных молекул, описывающая зависимость энергии взаимодействия двух частиц от расстояния между ними. Эта модель достаточно реалистично передаёт свойства реального взаимодействия сферических неполярных молекул и поэтому широко используется в расчётах и при компьютерном моделировании. Здесь r — расстояние между центрами частиц; ε — глубина потенциальной ямы; σ — расстояние, на котором энергия взаимодействия становится равной нулю. Параметры ε и σ являются характеристиками атомов соответствующего вещества.

F07. Потенциал Букингема. Функция потенциальной энергии, предложенная Ричардом Букингемом в качестве приближения для энергии Ван-дер-Ваальсовых сил и альтернативная потенциалу Леннарда — Джонса F06.

F08. Потенциал Морзе. Функция потенциальной энергии электростатического поля, предложенная Филиппом Морзе как приближение для энергии двухатомной молекулы. Здесь r_e — равновесное расстояние, а D_e — глубина потенциальной ямы. Параметр a определяет "ширину" потенциальной ямы, и может быть выражен через коэффициент упругости k_e и D_e следующим образом

$$a = \sqrt{\frac{k_e}{2D_e}}. \tag{7}$$

F09. Функция распределения Максвелла — Больцмана (по скорости). Применяется в кинетической теории газов и физике плазмы, представляя собой распределение тепловых скоростей частиц массы m при температуре T. Величина

$$v_T = \sqrt{\frac{kT}{m}} \tag{8}$$

имеет смысл характерной тепловой скорости.

F10. Псевдопотенциал Юкавы (экранированный кулоновский потенциал). Параметры имеют следующий смысл: q — заряд, λ_D — дебаевский радиус экранирования. Применяется для описания экранированных взаимодействий в плазме и электролитах, а также в ядерной физике (обмен мезонами).

5.2.5 Расчёт динамики материальной точки (P5)

Данное задание составлено на основе раздела *Расчёт Д-1* пособия [59].

Описание задания. Цель расчёта — приобретение опыта построения расчётной механической модели по описанию задачи; освоение методики составления дифференциальных уравнений движения выбранной модели — материальной точки; знакомство с методами аналитического и численного исследования уравнений. Требуется аналитически найти установившееся движение и оценить характерное время переходного процесса. Полученные оценки должны быть использованы для для выбора интервала интегрирования при численном анализе уравнений. Основная задача для расчёта на компьютере — определение переходного процесса выхода системы на установившийся режим при заданных начальных условиях.

Варианты заданий представлены на рис. 1 – 3. В описании каждого задания на рис. *а* схематически изображён исследуемый объект, на рис. *б* — его расчётная механическая модель. В качестве модели рассматривается материальная точка M, совершающая плоское движение. Моделью определяются силы следующего вида: сила $\mathbf{F}_0$, приводящая точку в движение или тормозящая её; вес $\mathbf{G}$; разность архимедовой силы и веса, задаваемая в вариантах 2, 10, 12, 15, 21, 25 в долях веса; силы упругости $\mathbf{F}_1$, $\mathbf{F}_2$; силы вязкого трения $\mathbf{S}_1$, $\mathbf{S}_2$.

Силы упругости $\mathbf{F}_{1,2}$ пропорциональны с коэффициентами $-k_{1,2}$ векторам деформаций соответствующих пружин модели, указанных на рисунках. В зависимости от варианта деформации пружин задаются на рисунках векторами $\mathbf{r}_{AM}$, $\mathbf{r}_{BM}$, $\mathbf{r}_{OM}$. Деформируемые элементы модели безынерционные.

Силы упругости реализуются в исходной схеме рис. *а* пружинами амортизаторов; за счёт поперечной и продольной упругости тросов; силами системы воздушной подвески в варианте 4; силами упругости ледовой поверхности в варианте 14; архимедовыми силами при частичном погружении тела в жидкость. Прямая OO_1 на рисунках вариантов 1, 9, 14, 19, 24 указывает положения точки, для которых силы поперечной упругости обращаются в ноль.

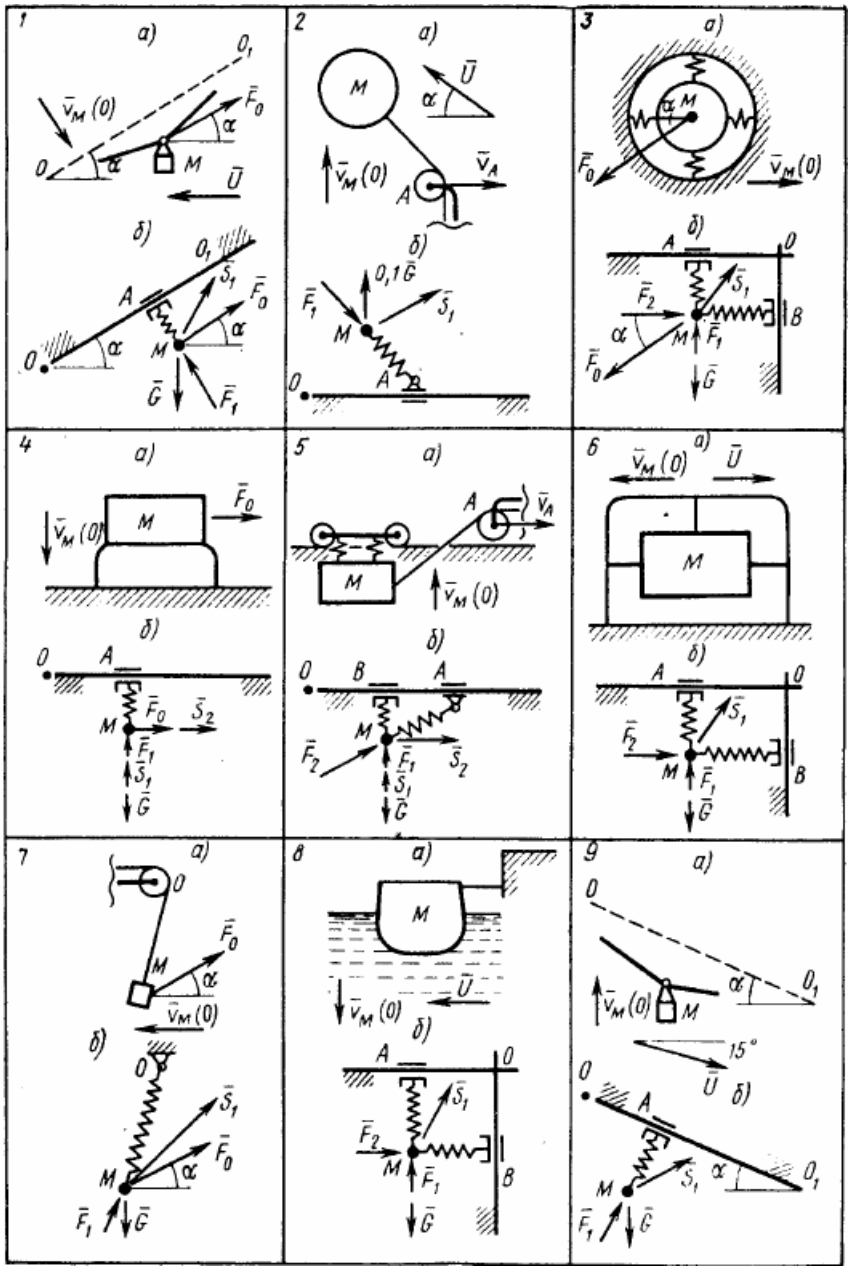

Рис. 1: Варианты 1 – 9 для задания Р5

В вариантах 3, 6, 8, 13, 17, 22, 26 предполагается, что перемещения точки M малы по сравнению с размерами элементов схемы на рис. *а*. При этом на рис. *б* можно принять, что направления линий действия упругих сил не зависят от деформаций.

В вариантах 2, 5, 7, 10, 12, 13, 15, 16, 18, 20, 21, 25, 27 принято упрощающее предположение о том, что вектор продольной деформации троса вместе с упругой силой нулю при совпадении точки M с блоком. Односторонний характер упругих сил, моделирующих продольную податливость тросов и архимедову силу, не обсуждается.

В большинстве вариантов сила вязкого трения пропорциональна с коэффициентом $-\mu_1$ вектору скорости движения точки относительно среды. В вариан-

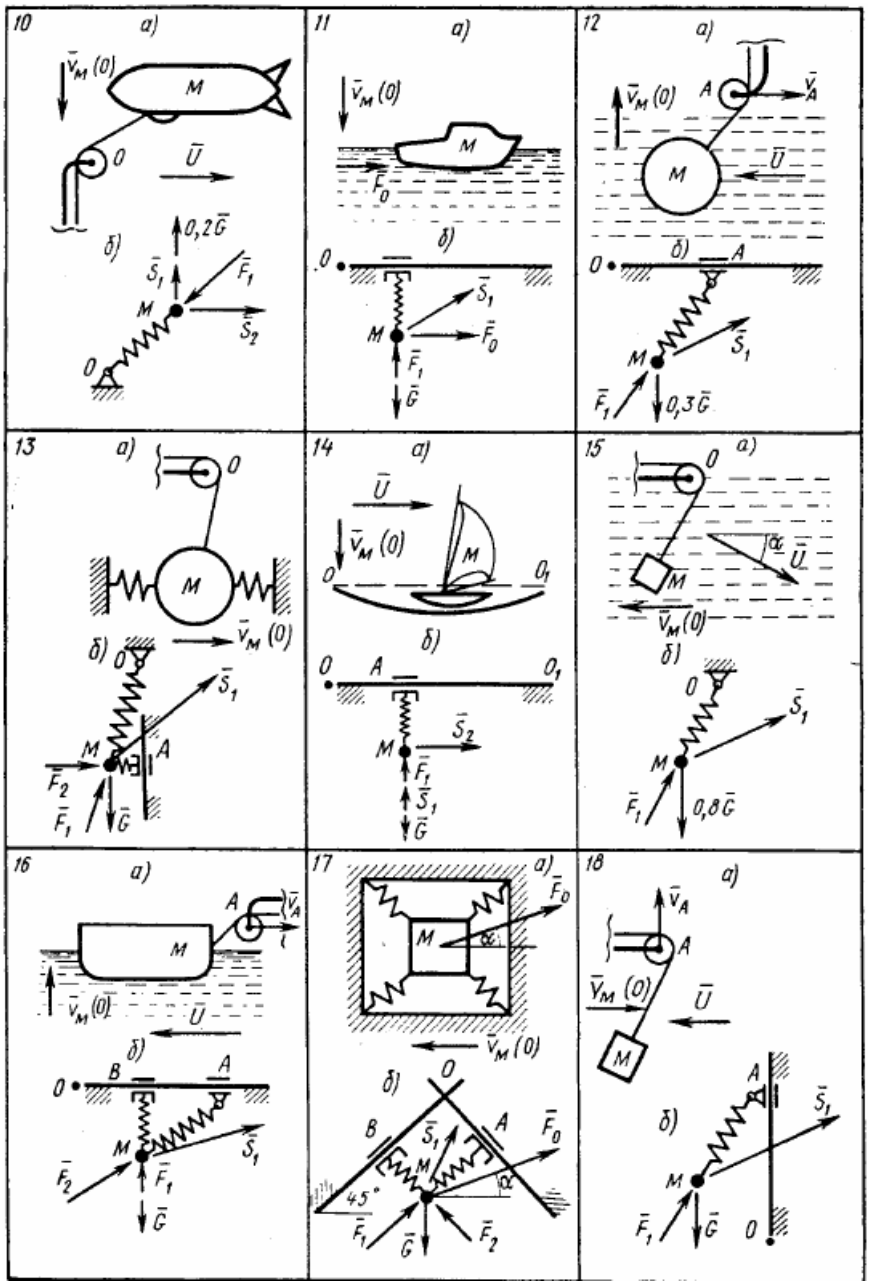

Рис. 2: Варианты 10 – 18 для задания Р5

тах 4, 5, 10, 14, 23, 25, 27 силы вязкого трения в вертикальном и горизонтальном направлениях пропорциональны с коэффициентами $-\mu_1$, $-\mu_2$ составляющим относительной скорости по этим направлениям.

На рис. *а* через **U** обозначен вектор скорости среды, в которой движется точка M, через $\boldsymbol{v}_A$ обозначена скорость точки A, заданная по условию для вариантов 2, 5, 12, 16, 18, 25, 26, 27. Для этих вариантов следует считать, что в начальный момент времени точка A совпадает с указанной на рисунках неподвижной точкой O. На рис. *а* вектор $\boldsymbol{v}_M(0)$ указывает направление начальной скорости точки M. Его модуль даётся Таблицей 6. Задание начального положения точки M оговорено ниже.

Числовые значения параметров определяются формулами (9) и Таблицей 6.

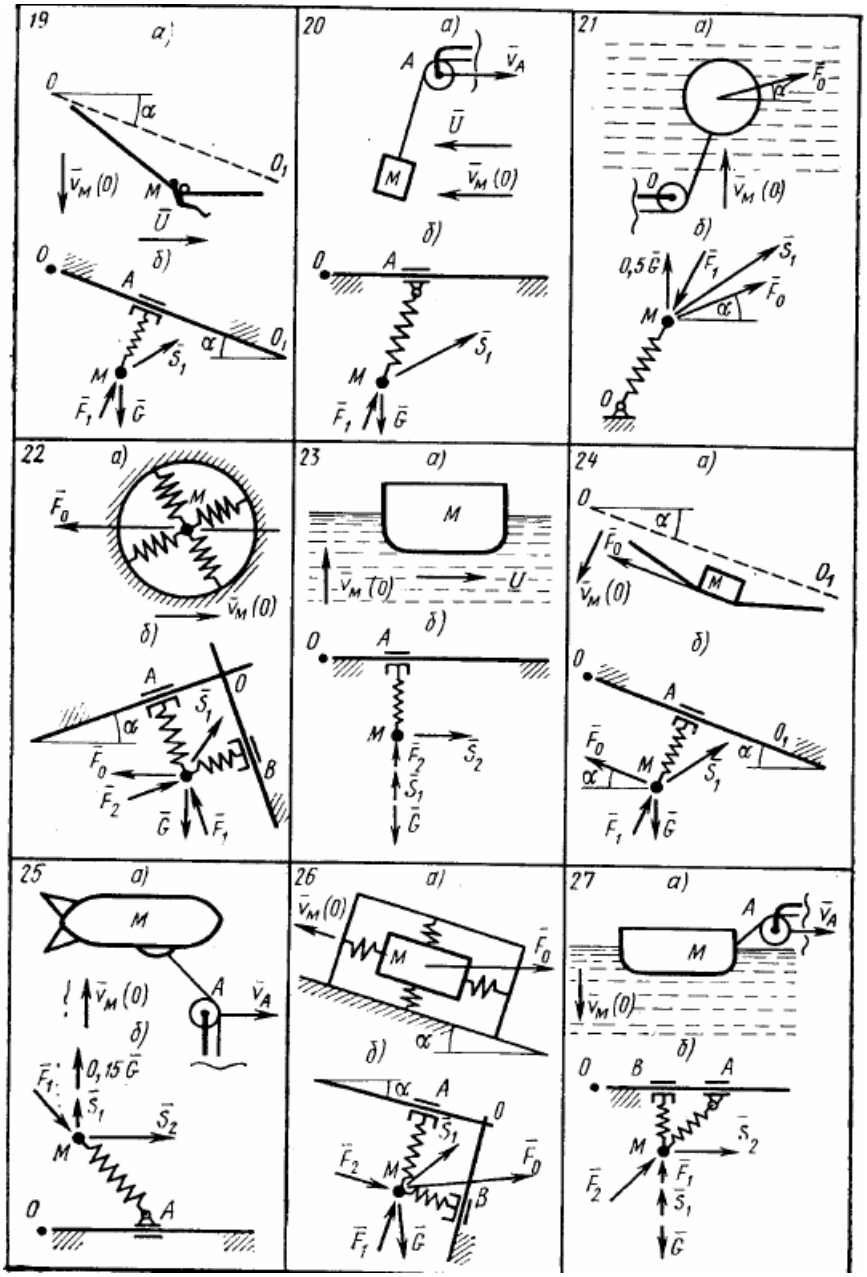

Рис. 3: Варианты 19 – 27 для задания Р5

Часть данных приведена на рисунках.

$$m = m_{\text{т}}(1 + 0.01N); \quad k_1 = k_{1\text{т}}(1 + 0.01n). \tag{9}$$

Здесь N — это номер группы (группе 325 соответствует $N = 1$, а 825 — $N = 2$), а n — это последняя цифра номера студенческого билета. Нижний индекс “т” означает, что соответствующее значение берётся из Таблицы 6.

Требуется исследовать движение материальной точки. Перечень пунктов исследования оговорён ниже в примере. Содержание задач следующее:

1. Подъём заданной силой вагонетки подвесной дороги.
2. Буксировка аэростата с заданной скоростью.

Таблица 6: Числовые параметры для задания Р5 (п. 5.2.5)

Вариант	$m_т$	$k_{1т}$	k_2	μ_1	μ_2	α, °	F_0	v_A	U	γ	$v_M(0)$
1	$1.3 \cdot 10^3$	$1.2 \cdot 10^4$	–	$1.4 \cdot 10^3$	–	30	$6 \cdot 10^3$	–	2.5	0.15	1.8
2	$0.8 \cdot 10^3$	$1.1 \cdot 10^3$	–	$0.28 \cdot 10^3$	–	60	–	5	3.2	0.5	0.9
3	$2.5 \cdot 10^{-4}$	150	150	$8 \cdot 10^{-2}$	–	45	$1.8 \cdot 10^{-2}$	–	–	0.3	$7 \cdot 10^{-2}$
4	$12 \cdot 10^3$	$1.4 \cdot 10^6$	–	$1.7 \cdot 10^5$	$0.9 \cdot 10^4$	–	$4.8 \cdot 10^4$	–	–	0.17	0.7
5	$6 \cdot 10^3$	$3.6 \cdot 10^5$	$1.2 \cdot 10^5$	$6 \cdot 10^4$	$1.5 \cdot 10^4$	–	–	4.2	–	0.22	1.6
6	$22 \cdot 10^3$	$1.9 \cdot 10^6$	$0.45 \cdot 10^6$	$0.8 \cdot 10^5$	–	–	–	–	3	0.1	0.25
7	$1.5 \cdot 10^2$	$1.7 \cdot 10^3$	–	$2.6 \cdot 10^2$	–	20	$2.8 \cdot 10^2$	–	–	0.13	2.2
8	$20 \cdot 10^3$	$4.3 \cdot 10^5$	$1.6 \cdot 10^6$	$4.1 \cdot 10^4$	–	–	–	–	2.6	0.31	0.23
9	$5 \cdot 10^2$	$6.1 \cdot 10^3$	–	$4.3 \cdot 10^2$	–	10	–	–	3	0.14	1.8
10	$65 \cdot 10^3$	$1.3 \cdot 10^5$	–	$5.8 \cdot 10^4$	$1.7 \cdot 10^4$	–	–	–	5	0.24	1.7
11	$1.5 \cdot 10^3$	$4.5 \cdot 10^3$	–	$1.6 \cdot 10^3$	–	–	$3.2 \cdot 10^3$	–	–	0.12	1.3
12	$5.5 \cdot 10^3$	$8 \cdot 10^4$	–	$5.3 \cdot 10^3$	–	–	–	2	3	0.33	0.6
13	$2.4 \cdot 10^3$	$2.3 \cdot 10^5$	$8 \cdot 10^4$	$8.3 \cdot 10^3$	–	–	–	–	–	0.27	0.9
14	$3.5 \cdot 10^2$	$7.1 \cdot 10^3$	–	$1.4 \cdot 10^3$	$2.1 \cdot 10^2$	–	–	–	5	0.11	1.4
15	$9 \cdot 10^3$	$1.1 \cdot 10^5$	–	$1.9 \cdot 10^4$	–	30	–	–	8	0.42	1.2
16	$50 \cdot 10^3$	$2.8 \cdot 10^5$	$6.2 \cdot 10^5$	$7 \cdot 10^4$	–	–	–	4	3	0.15	0.7
17	9	$8 \cdot 10^2$	$1.1 \cdot 10^3$	$0.8 \cdot 10^2$	–	25	120	–	–	0.22	0.9
18	$3 \cdot 10^3$	$6.2 \cdot 10^4$	–	$4.5 \cdot 10^3$	–	–	–	2	4	0.35	1.3
19	80	$6.5 \cdot 10^2$	–	75	–	15	–	–	2	0.12	1.5
20	$5 \cdot 10^3$	$5.3 \cdot 10^4$	–	$6.1 \cdot 10^3$	–	–	–	2.5	3.3	0.45	1.4
21	$1.2 \cdot 10^2$	$6.2 \cdot 10^3$	–	$3.8 \cdot 10^2$	–	40	$4.2 \cdot 10^2$	–	–	0.28	0.7
22	$4.8 \cdot 10^{-4}$	80	80	0.2	–	35	$1.3 \cdot 10^{-2}$	–	–	0.11	$3 \cdot 10^{-2}$
23	$150 \cdot 10^3$	$6.3 \cdot 10^5$	–	$1.2 \cdot 10^5$	$3.5 \cdot 10^4$	–	–	–	2.5	0.4	2.3
24	60	$7.2 \cdot 10^2$	–	80	–	15	45	–	–	0.16	1.5
25	$70 \cdot 10^3$	$9 \cdot 10^4$	–	$8.2 \cdot 10^4$	$1.5 \cdot 10^4$	–	–	3	–	0.35	0.9
26	$8.5 \cdot 10^3$	$8.8 \cdot 10^5$	$2.5 \cdot 10^5$	$3.8 \cdot 10^4$	–	10	$3.5 \cdot 10^4$	–	–	0.24	0.7
27	$270 \cdot 10^3$	$1.3 \cdot 10^6$	$0.45 \cdot 10^6$	$1.8 \cdot 10^5$	$0.4 \cdot 10^5$	–	–	4.5	–	0.5	2.3

3. Воздействие заданной силы на чувствительную массу акселерометра.
4. Разгон платформы на воздушной подушке заданной силой.
5. Буксировка амортизированного груза с заданной скоростью.
6. Ветровое возмущение удерживаемого тросами контейнера.
7. Возмущение висящего на тросе груза заданной силой.
8. Возмущение потоком удерживаемого тросом понтона.
9. Свободный спуск вагонетки подвесной дороги.
10. Ветровое возмущение дирижабля у причальной мачты.
11. Разгон катера заданной силой.
12. Буксировка с заданной скоростью подводного аппарата.
13. Свободные колебания висящего груза с боковыми амортизаторами.
14. Разгон буера ветром.
15. Возмущение потоком висящего груза.

16. Буксировка баржи с заданной скоростью.
17. Воздействие заданной силы на прибор в пружинном амортизаторе.
18. Подъём груза краном с заданной скоростью.
19. Свободный спуск альпиниста по канатной дороге.
20. Перемещение груза краном с заданной скоростью.
21. Воздействие заданной силы на буй.
22. Воздействие заданной силы на чувствительную массу акселерометра.
23. Свободный дрейф судна в потоке.
24. Скольжение груза с трением по ленте транспортёра.
25. Буксировка дирижабля с заданной скоростью.
26. Воздействие заданной силы на контейнер в пружинном подвесе.
27. Буксировка судна с заданной скоростью.

Указания к составлению уравнений движения. Движение точки под действием сил $\mathbf{F}_1$, $\mathbf{S}_1$, ... описывается уравнением Ньютона в векторной форме

$$m\frac{\mathrm{d}\boldsymbol{v}_M}{\mathrm{d}t} = \mathbf{F}_1 + \mathbf{S}_1 + \dots\,. \tag{10}$$

Зависимость векторов сил упругости от векторов деформации даётся выражениями вида

$$\mathbf{F}_1 = -k_1\mathbf{r}_{AM},\ \dots\,. \tag{11}$$

Если точка находится в среде, которая движется с заданной скоростью $\mathbf{U}$, то сила вязкого трения зависит от скорости точки относительно среды:

$$\mathbf{S}_1 = -\mu_1(\boldsymbol{v}_M - \mathbf{U}). \tag{12}$$

Векторы деформации $\mathbf{r}_{AM}$, ... в (10) следует выразить через радиус-векторы $\mathbf{r}_M$, $\mathbf{r}_A$ точек M, A, ..., выбрав точку O, указанную на рисунках, за начало векторов:

$$\mathbf{r}_{AM} = \mathbf{r}_M - \mathbf{r}_A, \dots \tag{13}$$

В вариантах с заданным движением точки A вектор $\mathbf{r}_A$ зависит от времени.

Выберем систему координат Oxy, в которой векторы из (11), (12), (13), разлагаются наиболее простым образом:

$$\begin{aligned} \mathbf{r}_M &= x\,\mathbf{i} + y\,\mathbf{j}; \qquad \mathbf{r}_A = x_A\,\mathbf{i} + y_A\,\mathbf{j}; \\ \boldsymbol{v}_M &= \dot{x}\,\mathbf{i} + \dot{y}\,\mathbf{j}; \qquad \mathbf{U} = U_x\,\mathbf{i} + U_y\,\mathbf{j}. \end{aligned} \tag{14}$$

Подстановка (14) в (11), (12), (13) и далее в (10) даст два однотипных уравнения вида:

$$m\ddot{x} = -k(x - x_A) - \mu(\dot{x} - U_x) + G_x + \ldots, \ldots \tag{15}$$

Здесь через k и μ обозначены коэффициенты упругости и вязкого трения, приведённые к координате x.

Определение установившегося движения системы и характерных времён переходных процессов. Установившиеся движения системы описываются решениями следующего вида:

$$\begin{aligned} x^* &= C_1 t + C_2; \qquad & y^* &= C_3 t + C_4; \\ \dot{x}^* &= v^*_{Mx} = C_1; \qquad & \dot{y}^* &= v^*_{My} = C_3. \end{aligned} \tag{16}$$

Здесь C_1, C_2, C_3, C_4 — константы, определяемые подстановкой (16) в (15).

Определение начальных условий. Начальные условия системы (15) по координатам задаются Таблицей 6 в рассогласованиях[2] от установившегося решения

$$\frac{\mathbf{r}_M(0)}{\mathbf{r}^*_M(0)} = \gamma. \tag{17}$$

Используя (16), из (17) получаем

$$x(0) = \gamma C_2; \quad y(0) = \gamma C_4. \tag{18}$$

В вариантах 1, 4, 9, 11, 14, 19, 23, 24, уравнения которых не зависят от переменной x, следует принять $C_2 = 0$, т. е. $x(0) = 0$. Этого можно добиться за счёт сдвига начала отсчёта вдоль оси x. Начальные скорости даются в описаниях вариантов и Таблице 6.

Определение характерного времени движения. Система (15) состоит из двух независимых уравнений. Запишем характеристическое уравнение, которое соответствует первому уравнению из (15), описывающему движение по переменной x:

$$m\lambda^2 + \mu\lambda + k = 0. \tag{19}$$

[2]Не следует понимать это буквально как деление векторов. Смысл этой записи виден из выражения (18).

Уравнение (19) имеет либо комплексные сопряжённые корни

$$\lambda_{1,2} = -\delta \pm i\omega; \quad \delta > 0,\ \omega > 0, \tag{20}$$

либо вещественные отрицательные

$$\lambda_{1,2} = -\delta_{1,2}; \quad \delta_{1,2} > 0. \tag{21}$$

В случае (20) решение системы (15) имеет колебательные затухающие составляющие с циклической частотой ω. Период этих колебаний $T_{\text{osc}} = 2\pi/\omega$. Следовательно, характерным временем, на котором выявляется колебательный характер движения, будет время порядка T_{osc}.

В случае (21) решение системы (15) имеет экспоненциальные составляющие вида $Ce^{-\delta_{1,2}t}$ или $Ce^{-t/T_{1,2}}$. Здесь $T_{1,2} = 1/\delta_{1,2}$ — постоянные времени апериодического процесса. Характерным временем, на котором выявляется апериодический характер каждой из этих составляющих, будет время порядка $T_{1,2}$.

Характерные времена движения системы по переменной y определяются аналогично в силу второго уравнения системы (15).

Указания к решению задач на компьютере. Система дифференциальных уравнений (15) приводится к форме Коши и интегрируется численным методом на интервале τ. Чтобы при счёте выявилось поведение всех составляющих движения, величину τ следует выбрать равной наибольшему из характерных времён этих составляющих.

На печать выводятся переменные t, x, y, $v_{Mx} = \dot{x}$, $v_{My} = \dot{y}$ с шагом, равным $1/120$[3].

Один из возможных вариантов программы с использованием конечно-разностной схемы Эйлера приведён в примере, рассмотренном в п. 5.2.6. Студентам рекомендуется интегрировать уравнения движения методом Рунге — Кутты.

Контроль решения. Числовые значения параметров подобраны так, что характерные времена всех составляющих движения отличаются не более чем в 2 – 3 раза. Бóльшие различия указывают на ошибку вычисления корней характеристического уравнения.

[3]Как вы думаете, почему предлагается именно такой шаг? *Подсказка:* поищите информацию о таком понятии, как кадровая частота.

Численный счёт должен дать движение с составляющими такого вида, какие получились при анализе корней характеристического уравнения. Переходный процесс, вычисленный на компьютере, должен стремиться к установившемуся движению (16), найденному аналитически.

Защита расчёта. Типовые вопросы:

1. Объяснить переход от схемы объекта к расчётной модели.
2. Объяснить запись выражения для скорости точки относительно потока.
3. Записать уравнения движения для изменённого набора сил, действующих на точку.
4. Объяснить методику составления характеристического уравнения системы.
5. Объяснить смысл характерного времени для каждой составляющей движения.
6. Каким будет движение, если рассогласования начальных условий от установившегося движения будут равны нулю?
7. По графику $x(t)$ найти значение производной в заданный момент времени и сравнить с результатом счёта.
8. Для заданного момента времени построить на траектории вектор скорости и ускорения точки.
9. По графику $v_{Mx}(t)$ приближённо построить график $x(t)$. Сравнить с результатами счёта.
10. По графику $v_{Mx}(t)$ приближённо построить график $a_{Mx}(t)$.
11. Объяснить алгоритм работы программы.

5.2.6 Пример выполнения задания Р5

(Вариант 31, $n = 5$, $N = 3$)

Постановка задачи. Груз M поднимается в потоке жидкости по несущему тросу 1, который в ненагруженном состоянии совпадает с прямой OO_1 (рис. 4, *а*). Груз приводится в движение тележкой A, соединённой с грузом тяговым тросом 2. Тележка движется по прямой OO_1 с заданной скоростью $\boldsymbol{v}_A$. Учитывается поперечная упругость троса 1 и продольная — троса 2. Масса тросов считается равной нулю. В начальный момент времени точка A совпадает с

заданной точкой O. Скорость потока $\mathbf{U}$ горизонтальна.

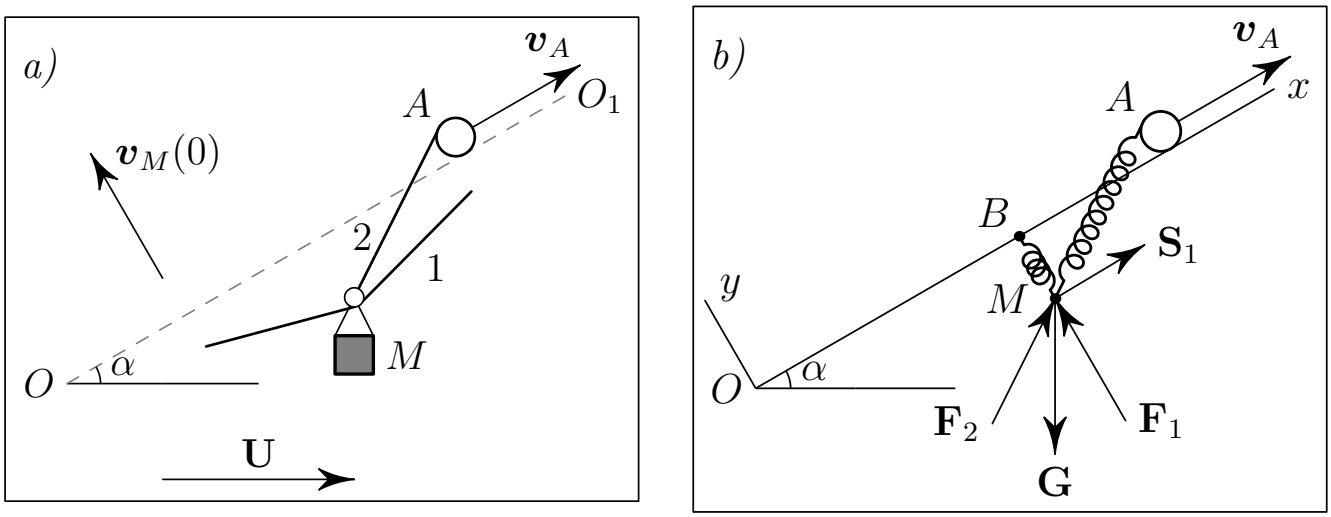

Рис. 4: Вариант 31 для задания Р5

Дано: $m_\text{т} = 1\,000$ кг; $k_{1\text{т}} = 10\,000$ Н/м; $k_2 = 2\,500$ Н/м; $\mu_1 = 3\,800$ Н $\cdot$ с/м; $\alpha = 30°$; $v_A = 1$ м/с; $U = 0.7$ м/с; $\gamma = 0.5$; $v_M(0) = 1$ м/с.

Требуется: **1.** Составить дифференциальное уравнение движения точки. **2.** Найти установившееся движение. **3.** Определить характерные времена, за которые развиваются составляющие движения. **4.** Проинтегрировать на компьютере уравнения движения, найти переходный процесс выхода на установившееся движение. **5.** Построить графики $v_{Mx}(t)$, $v_{My}(t)$, $x(t)$, $y(t)$ и траекторию движения точки в плоскости Oxy.

Составление уравнений движения. Расчётная механическая модель системы представлена на рис. 4, *б*. На материальную точку M действует сила тяжести $\mathbf{G}$, силы упругости $\mathbf{F}_{1,2}$ тросов 1 и 2, сила вязкого трения $\mathbf{S}_1$.

Запишем в векторной форме закон Ньютона для движения точки:

$$m\frac{\mathrm{d}\boldsymbol{v}_M}{\mathrm{d}t} = \mathbf{G} + \mathbf{F}_1 + \mathbf{F}_2 + \mathbf{S}_1. \tag{22}$$

Здесь

$$\mathbf{F}_1 = -k_1\mathbf{r}_{BM}, \quad \mathbf{F}_2 = -k_2\mathbf{r}_{AM}, \quad \mathbf{S}_1 = -\mu\boldsymbol{v}_r. \tag{23}$$

Векторы поперечной деформации $\mathbf{r}_{BM}$ троса 1, продольной деформации $\mathbf{r}_{AM}$ троса 2 и вектор $\boldsymbol{v}_r$ скорости относительно потока определяются выражениями

$$\mathbf{r}_{BM} = \mathbf{r}_M - \mathbf{r}_B, \quad \mathbf{r}_{AM} = \mathbf{r}_M - \mathbf{r}_A; \quad \boldsymbol{v}_r = \boldsymbol{v}_M - \mathbf{U}. \tag{24}$$

Уравнение (22) удобно расписать в проекциях на оси системы коорди-

нат Oxy с началом в точке O и осью x, совпадающей с направлением OO_1. В этой системе координат векторы из (22), (23), (24) задаются проекциями

$$\begin{aligned}
&\mathbf{r}_M(x,y), \quad \mathbf{r}_B(x,0), \quad \mathbf{r}_A(v_A t,0);\\
&\boldsymbol{v}_M(\dot{x},\dot{y}), \quad \mathbf{U}(U\cos\alpha,-U\sin\alpha);\\
&\mathbf{G}(-G\sin\alpha,-G\cos\alpha); \quad \mathbf{F}_1(0,-k_1 y);\\
&\mathbf{F}_2(-k_2(x-v_A t),-k_2 y);\\
&\mathbf{S}_1(-\mu_1(\dot{x}-U\cos\alpha),-\mu_1(\dot{y}+U\sin\alpha)).
\end{aligned} \tag{25}$$

Подставив (25) в (22), получим

$$\begin{aligned}
m\ddot{x} &= -k_2(x-v_A t)-\mu_1(\dot{x}-U\cos\alpha)-G\sin\alpha;\\
m\ddot{y} &= -(k_1+k_2)y-\mu_1(\dot{y}+U\sin\alpha)-G\cos\alpha.
\end{aligned} \tag{26}$$

Для заданных числовых значений параметров (26) примет вид

$$\begin{aligned}
\ddot{x} &= -2.43(x-t)-3.69\dot{x}-2.69;\\
\ddot{y} &= -12.62y-3.69\dot{y}-9.83.
\end{aligned} \tag{27}$$

Определение установившегося движения. Отыскиваем частные решения системы (27) вида

$$x^* = C_1 t + C_2, \quad y^* = C_3 t + C_4, \tag{28}$$

где C_1, C_2, C_3, C_4 — неизвестные константы. Подставим (28) в (27) и сгруппируем слагаемые:

$$\begin{aligned}
2.43(C_1-1)t+(2.43C_2+3.69C_1+2.69)&=0;\\
12.62C_3 t+(3.69C_3+12.62C_4+9.83)&=0.
\end{aligned} \tag{29}$$

Уравнения (29) обращаются в тождества, когда коэффициенты при первой и нулевой степени t порознь равны нулю. Отсюда

$$C_1 = 1; \quad C_2 = -2.63; \quad C_3 = 0; \quad C_4 = -0.78. \tag{30}$$

Установившееся движение (28) по координатам примет вид

$$x^* = t - 2.63; \quad y^* = -0.78, \tag{31}$$

по скоростям

$$v^*_{Mx} = \frac{\mathrm{d}}{\mathrm{d}t}x^* = 1; \quad v^*_{My} = \frac{\mathrm{d}}{\mathrm{d}t}y^* = 0. \tag{32}$$

Определение начальных условий. Начальные условия системы (27) по координатам заданы в рассогласованиях от установившегося решения

$$\mathbf{r}_M(0) = 0.5\mathbf{r}^*_M(0). \tag{33}$$

Из (31), (33) запишем

$$\begin{aligned} x(0) &= 0.5x^*(0) = -1.32 \text{ м}; \\ y(0) &= 0.5y^*(0) = -0.39 \text{ м}. \end{aligned} \tag{34}$$

Оценка характерного времени переходного процесса. Запишем характеристические уравнения, соответствующие системе (27):

$$\begin{aligned} \lambda^2 + 3.69\lambda + 2.43 = 0, \\ \lambda^2 + 3.69\lambda + 12.62 = 0. \end{aligned} \tag{35}$$

Корни этих уравнений[4] $\lambda_1 = -2.83$, $\lambda_2 = -0.86$, $\lambda_{3,4} = -1.84 \pm i \cdot 3.04$. Следовательно, движение системы имеет одну колебательную и две апериодические составляющие. Найдём период и постоянные времени этих составляющих:

$$T_{\text{osc}} = \frac{2\pi}{3.04} = 2.07 \text{ с}; \quad T_1 = \frac{1}{2.83} = 0.35 \text{ с}; \quad T_2 = \frac{1}{0.86} = 1.16 \text{ с}. \tag{36}$$

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <math.h>
4
5  typedef struct coeff {
6      double a, b, c;
7  } coeff;
8
9  typedef struct result {
10     double x1, x2, y1, y2;
11 } result;
12
```

[4]Исходный текст программы на языке C, решающей задачу поиска комплексных корней квадратного уравнения приведён в Листинге 1.

```
13 int input(int, char**, coeff*);
14 void help(char**);
15 result calc(coeff);
16 void fixzero(result*);
17 void print(result);
18
19 int main(int argc, char ** argv) {
20     coeff C;
21     if ( input(argc, argv, &C) ) {
22         help(argv);
23         exit(EXIT_FAILURE);
24     }
25     result Ans = calc(C);
26     print(Ans);
27     return 0;
28 }
29
30 int input(int argc, char ** argv, coeff * C) {
31     if (argc != 4)
32         return 1;
33     char * p;
34     C->a = strtod(argv[1], &p);
35     if (argv[1] == p || *p != '\0' || p == NULL || C->a == 0)
36         return 1;
37     C->b = strtod(argv[2], &p);
38     if (argv[2] == p || *p != '\0' || p == NULL)
39         return 1;
40     C->c = strtod(argv[3], &p);
41     if (argv[3] == p || *p != '\0' || p == NULL)
42         return 1;
43     return 0;
44 }
45
46 void help(char ** argv) {
47     printf("Error!\nProgram usage:\n\t%s a b c\n", argv[0]);
48     return;
49 }
50
51 result calc(coeff C) {
52     result A;
53     double D = C.b * C.b - 4*C.a*C.c;
54     if ( D < 0 ) {
55         A.x1 = -C.b/2/C.a;
```

```
56          A.x2 = -sqrt(-D)/2/C.a;
57          A.y1 = -C.b/2/C.a;
58          A.y2 = sqrt(-D)/2/C.a;
59      }
60      if ( D == 0) {
61          A.x1 = -C.b/2/C.a;
62          A.x2 = 0;
63          A.y1 = -C.b/2/C.a;
64          A.y2 = 0;
65      }
66      if ( D > 0) {
67          A.x1 = -(C.b + sqrt(D))/2/C.a;
68          A.x2 = 0;
69          A.y1 = (-C.b + sqrt(D))/2/C.a;
70          A.y2 = 0;
71      }
72      fixzero(&A);
73      return A;
74 }
75
76 void fixzero(result * A) {
77      if ( A->x1 == 0 && signbit(A->x1) ) A->x1 = +0.;
78      if ( A->x2 == 0 && signbit(A->x2) ) A->x2 = +0.;
79      if ( A->y1 == 0 && signbit(A->y1) ) A->y1 = +0.;
80      if ( A->y2 == 0 && signbit(A->y2) ) A->y2 = +0.;
81      return;
82 }
83
84 void print(result A) {
85      printf("x1 = (%g, %g)\nx2 = (%g, %g)\n", A.x1, A.x2, A.y1, A.y2
            );
86 }
```

Листинг 1: Программа поиска комплексных корней квадратного уравнения

Решение задачи на компьютере и обработка результатов. Приведём систему (27) к форме Коши:

$$\begin{aligned} \dot{x} &= v_{Mx}, \\ \dot{y} &= v_{My}, \\ \dot{v}_{Mx} &= a_{Mx}, \\ \dot{v}_{My} &= a_{My}, \\ a_{Mx} &= -2.43(x - t) - 3.69v_{Mx} - 2.69, \\ a_{My} &= -12.62y - 3.69v_{My} - 9.83. \end{aligned} \tag{37}$$

Систему уравнений (37) решаем на компьютере с помощью программы, исходный текст которой приведён в Листинге 2. Для интегрирования используем конечно-разностную схему Эйлера. В качестве интервала интегрирования выберем наибольшую из величин в (36). Тогда $\tau = T_{\text{osc}} = 2.07$ с. За шаг интегрирования примем $\Delta t = \tau/1\,200 \approx 1.72$ мс. Шаг печати равен $10\Delta t = 17.2$ мс. По результатам счёта построены графики $x(t)$, $y(t)$, $v_{Mx}(t)$, $v_{My}(t)$ и $y(x)$ на рис. 5. Примеры анимации, созданные по результатам расчёта, можно увидеть в следующих файлах: P5-example.gif и P5-example-AM.gif.

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main(int argc, char ** argv) {
5      if (argc != 2) {
6          printf("Error!\nProgram usage:\n\t%s output\n", argv[0]);
7          exit(EXIT_FAILURE);
8      }
9      const double dt = 1.72e-3;
10     double t  = 0;
11     double x  = -1.32;
12     double y  = -0.39;
13     double vx = 0;
14     double vy = 1;
15     double ax, ay;
16     const int N = 1200;
17     int i = 0;
18     FILE *fp = fopen(argv[1], "w");
19     if (!fp) {
20         perror("File open error");
21         exit(EXIT_FAILURE);
```

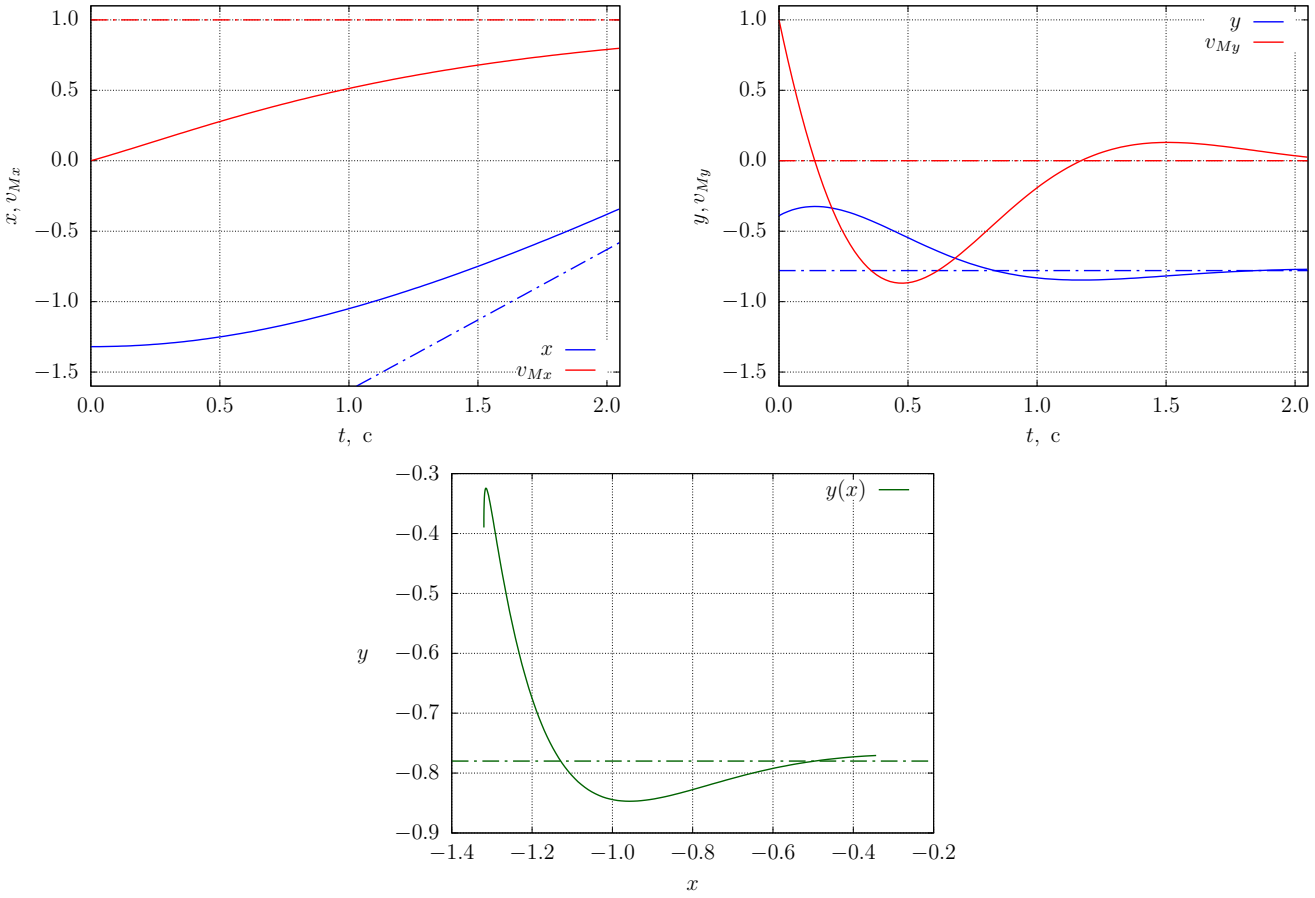

Рис. 5: Результаты численного интегрирования системы (37)

```
22      }
23      fprintf(fp, "# t\tx\ty\tvx\tvy\tax\tay\n");
24      while (i < N) {
25          ax = -2.43 * (x - t) - 3.69 * vx - 2.69;
26          ay = -12.62 * y - 3.69 * vy - 9.83;
27          if (i%10 == 0)
28              fprintf(fp, "%g\t%g\t%g\t%g\t%g\t%g\t%g\n", t, x, y, vx
                    , vy, ax, ay);
29          x  += vx * dt;
30          y  += vy * dt;
31          vx += ax * dt;
32          vy += ay * dt;
33          t  += dt;
34          i++;
35      }
36      fclose(fp);
37      return 0;
38  }
```

Листинг 2: Реализация схемы Эйлера

Контроль решения. На рис. 5 штрих-пунктиром изображены установившиеся решения построенные по (31), (32). Видно, что решение системы (27), полученное на компьютере, стремится к установившемуся. Характерные времена переходного процесса близки к величинам, приведённым в (36).

5.3 Варианты заданий на практику

Номер варианта индивидуальных заданий на практику закрепляется за каждым студентом по согласованию с преподавателем. Полный перечень вариантов приведён в Таблице 7. Для расшифровки этой таблицы следует использовать данные приведённые в Таблицах 2 – 6. При возникновении любых возможных проблем обращайтесь к преподавателю.

Таблица 7: Варианты заданий на практику

Вариант	Реферат	Задача				
		P1	P2	P3	P4	P5
V01	1	T1+I1+M1	T4	1	F01	1
V02	2	T2+I2+M2	T5	2	F02	2
V03	3	T3+I3+M3	T6	3	F03	3
V04	4	T4+I1+M4	T1	4	F04	4
V05	5	T5+I2+M5	T2	5	F05	5
V06	6	T6+I3+M6	T3	6	F06	6
V07	7	T1+I1+M7	T4	7	F07	7
V08	8	T2+I2+M8	T5	8	F08	8
V09	9	T3+I3+M1	T6	9	F09	9
V10	10	T4+I1+M2	T1	10	F10	10
V11	11	T5+I2+M3	T2	1	F01	11
V12	12	T6+I3+M4	T3	2	F02	12
V13	13	T1+I2+M5	T5	3	F03	13
V14	14	T2+I3+M6	T6	4	F04	14
V15	15	T3+I1+M7	T1	5	F05	15
V16	16	T4+I2+M8	T2	6	F06	16
V17	17	T5+I3+M1	T3	7	F07	17
V18	18	T6+I1+M2	T4	8	F08	18
V19	19	T1+I2+M3	T5	9	F09	19
V20	20	T2+I3+M4	T6	10	F10	20
V21	21	T3+I1+M5	T1	1	F01	21
V22	22	T4+I2+M6	T2	2	F02	22
V23	23	T5+I3+M7	T3	3	F03	23
V24	24	T6+I1+M8	T4	4	F04	24
V25	25	T1+I2+M1	T6	5	F05	25

Список использованных источников

[1] Д.А. Плешев и Ф.Ф. Асадуллин. *Учебная, технологическая и преддипломная практика: методические указания*. Сыктывкар: СЛИ, 2021, с. 40.

[2] А.В. Столяров. *Программирование: введение в профессию*. 2-е изд., испр. и доп.: в 3 томах. Т. I. Азы программирования. М.: МАКС Пресс, 2021, с. 704. URL: http://stolyarov.info/books/programming_intro/e2.

[3] А.В. Столяров. *Программирование: введение в профессию*. 2-е изд., испр. и доп.: в 3 томах. Т. II. Системы и сети. М.: МАКС Пресс, 2021, с. 704. URL: http://stolyarov.info/books/programming_intro/e2.

[4] Ф.Ф. Владстон. *Теоретический минимум по Computer Science. Всё, что нужно программисту и разработчику*. СПб: Питер, 2018, с. 224.

[5] Ч. Петцольд. *Код. Тайный язык информатики*. М.: Манн, Иванов и Фербер, 2019, с. 448.

[6] Н. Вирт. *Алгоритмы и структуры данных*. М.: Мир, 1989, с. 360.

[7] Б. Керниган и Д. Ритчи. *Язык программирования Си*. 3-е изд., испр. СПб.: Невский Диалект, 2001, с. 352.

[8] Т.Х. Кормен и др. *Алгоритмы: построение и анализ*. М.: И. Д. "Вильямс", 2013, с. 1328.

[9] Э.С. Реймонд. *Искусство программирования для Unix*. М.: И. Д. "Вильямс", 2005, с. 544. URL: http://catb.org/~esr/writings/taoup/.

[10] Дж. Фуско. *Linux. Руководство программиста*. СПб.: Питер, 2011, с. 448.

[11] roadmap.sh. *Computer Science Roadmap*. URL: https://roadmap.sh/computer-science.

[12] *Google Академия*. URL: https://scholar.google.com/.

[13] *Mendeley*. URL: https://www.mendeley.com.

[14] *arXiv.org*. URL: https://arxiv.org/.

[15] *Научная электронная библиотека*. URL: https://www.elibrary.ru/.

[16] *Web of Science*. URL: https://webofknowledge.com.

[17] *Scopus*. URL: https://www.scopus.com.

[18] *ScienceDirect*. URL: https://www.sciencedirect.com.

[19] *Springer Link*. URL: https://link.springer.com/.

[20] *Электронная библиотека Института инженеров по электротехнике и электронике*. URL: https://ieeexplore.ieee.org.

[21] *ИСТИНА — Интеллектуальная Система Тематического Исследования НАукометрических данных*. URL: https://istina.msu.ru/.

[22] *MathSciNet*. URL: https://mathscinet.ams.org.

[23] *Проект Евклид*. URL: https://projecteuclid.org.

[24] *MathNet*. URL: http://www.mathnet.ru/.

[25] И.А. Котельников и П.З. Чеботарёв. *LaTeX по-русски*. 3-е изд., перераб. и доп. Новосибирск: Сибирский хронограф, 2004, с. 496.

[26] С.М. Львовский. *Набор и вёрстка в системе LaTeX*. 5-е изд., перераб. М.: МЦНМО, 2014, с. 400.

[27] А.В. Столяров. *Сверстай диплом красиво: LaTeX за три дня*. М.: МАКС Пресс, 2010, с. 100. URL: http://stolyarov.info/books/latex3days.

[28] Д.Э. Кнут. *Все про TeX*. М.: Издательский дом "Вильямс", 2003, с. 560.

[29] А.В. Костырка. *Краткий курс благородного набора*. URL: https://kostyrka.ru/main/ru/typesetting-and-typography-crash-course-by-kostyrka.

[30] D. Kirsch. *Detexify — LaTeX symbol recognition*. URL: http://detexify.kirelabs.org/classify.html.

[31] *Sublime Text*. URL: https://www.sublimetext.com.

[32] *TeX Live*. URL: https://tug.org/texlive/.

[33] *Overleaf*. URL: https://www.overleaf.com/.

[34] *BibLaTeX — Sophisticated Bibliographies in LaTeX*. URL: https://ctan.org/pkg/biblatex.

[35] *Biber — A BibTeX replacement for users of BibLaTeX*. URL: https://ctan.org/pkg/biber.

[36] A. Feder. *BibTeX*. URL: http://www.bibtex.org/.

[37] *JabRef*. URL: https://www.jabref.org/.

[38] *Страница загрузки Mendeley Desktop*. URL: https://www.mendeley.com/download-desktop-new/.

[39] *Gnuplot homepage*. URL: http://www.gnuplot.info/.

[40] *Official gnuplot documentation*. URL: http://www.gnuplot.info/documentation.html.

[41] *Demos for gnuplot*. URL: http://gnuplot.sourceforge.net/demo/.

[42] *Gnuplot и с чем его едят*. URL: https://habr.com/ru/company/ruvds/blog/517450/.

[43] *Asymptote homepage*. URL: https://asymptote.sourceforge.io/.

[44] *Asymptote documentation*. URL: https://asymptote.sourceforge.io/doc/index.html.

[45] Ю.М. Волченко. *Научная графика на языке Asymptote*. URL: http://math.volchenko.com/AsyMan.pdf.

[46] P. Ivaldi. *Euclidean geometry with Asymptote*. URL: http://www.piprime.fr/files/res/geometry_en.pdf.

[47] *Asymptote Gallery*. URL: https://asymptote.sourceforge.io/gallery/index.html.

[48] *Asymptote examples collection*. URL: https://asy.marris.fr/asymptote/.

[49] *Asymptote Web Application*. URL: http://asymptote.ualberta.ca/.

[50] А.В. Столяров. *Введение в язык Си++*. 5-е изд., испр. и доп. М.: МАКС Пресс, 2020, с. 156. URL: http://stolyarov.info/books/cppintro.

[51] *cplusplus.com*. URL: http://www.cplusplus.com.

[52] *GSL homepage*. URL: https://www.gnu.org/software/gsl/.

[53] *GCC, the GNU Compiler Collection*. URL: https://gcc.gnu.org/.

[54] *ROOT homepage*. URL: https://root.cern/.

[55] *GDB: The GNU Project Debugger*. URL: https://www.sourceware.org/gdb/.

[56] А.В. Столяров. *Архитектура ЭВМ и системное программное обеспечение. Пособие по выполнению лабораторных работ на ЭВМ в среде ОС UNIX*. М.: МГТУ ГА, 2009, с. 48. URL: http://stolyarov.info/books/unixref.

[57] freeCodeCamp. *Sorting Algorithms Explained with Examples in JavaScript, Python, Java, and C++*. URL: https://www.freecodecamp.org/news/sorting-algorithms-explained-with-examples-in-python-java-and-c/.

[58] А.С. Богатырёв. *Хрестоматия по программированию на Си в Unix*. М., 1995, с. 346. URL: https://www.opennet.ru/docs/RUS/bogatyrev/.

[59] И.В. Новожилов и М.Ф. Зацепин. *Типовые расчёты по теоретической механике на базе ЭВМ*. М.: Высш. шк., 1986, с. 136.