

Инженерная школа информационных технологий и робототехники  
 Направление подготовки: Прикладная математика и информатика  
 Отделение информационных технологий

### БАКАЛАВРСКАЯ РАБОТА

Тема работы
<b>Решение задачи квадратичного программирования на основе преобразования Гаусса-Жордана</b>

УДК 519.853.32:512.644

Студент

Группа	ФИО	Подпись	Дата
8Б51	Кулигин Сергей Михайлович		

Руководитель ВКР

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОМИ ШБИП	Вылегжанин Олег Николаевич	к.х.н., доцент		

### КОНСУЛЬТАНТЫ ПО РАЗДЕЛАМ:

По разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОСГН ШБИП	Подопрigора Игнат Валерьевич	к.э.н., доцент		

По разделу «Социальная ответственность»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Ассистент ООТД	Мезенцева Ирина Леонидовна			

### ДОПУСТИТЬ К ЗАЩИТЕ:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Руководитель ООП	Шевелев Геннадий Ефимович	к.ф.-м.н., доцент		

## ЗАПЛАНИРОВАННЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ ПО ООП

Код результатов	Результат обучения (выпускник должен быть готов)	Требования ФГОС, Критерии АИОР
<i>Профессиональные компетенции</i>		
<b>Р1</b>	Применять <i>глубокие математические и профессиональные знания</i> для решения задач научно-исследовательской, проектной, производственной и технологической деятельности в области системного и прикладного программирования.	Требования ФГОС (ОК-11, 12, ПК-3, 10), Критерий 5 АИОР (п. 5.2.1), согласованный с требованиями международных стандартов <i>EUR-ACE</i> и <i>FEANI</i> . Требования профессиональных стандартов Ассоциации предприятий компьютерных и информационных технологий Требования работодателей: ФГУП «РФЯЦ-ВНИИТФ им. академика Е.И. Забабахина», ООО НАЦ «Недра», ИХН СО РАН
<b>Р2</b>	Умение использовать знания по естественнонаучным дисциплинам при определении задач математического моделирования объектов и явлений в различных предметных областях	Требования ФГОС (ПК-3,9) Критерий 5 АИОР (п.5.2.3), согласованный с требованиями международных стандартов <i>EUR-ACE</i> и <i>FEANI</i> . Требования работодателей: ФГУП «РФЯЦ-ВНИИТФ им. академика Е.И. Забабахина», ООО «НАПО им. В.П.
<b>Р3</b>	Демонстрировать понимание сущности и значения информации в развитии современного общества, владение основными методами, способами и средствами получения, хранения, переработки информации; использование для решения коммуникативных задач современных технических средств и информационных технологий	Требования ФГОС (ОК-5, 11, 12,14,15, ПК-2, 6), Критерий 5 АИОР (п. 5.2.2), согласованный с требованиями международных стандартов <i>EUR-ACE</i> и <i>FEANI</i> Требования профессиональных стандартов Ассоциации предприятий компьютерных и информационных технологий Требования работодателей: Контек, ОАО «Газпром переработка», ООО Нижневартовскэнергогаз. Требования ФГОС (ОК-5, 11,

<b>P4</b>	Выполнять <i>инновационные</i> проекты с применением <i>глубоких профессиональных</i> знаний и <i>эффективных</i> методов проектирования для достижения <i>новых</i> результатов, обеспечивающих <i>конкурентные преимущества</i> в условиях экономических, экологических, социальных и других ограничений.	Требования ФГОС (ОК-14, ПК- 7, 9,14), Критерий 5 АИОР (п. 5.2.4), согласованный с требованиями международных стандартов <i>EUR-ACE</i> и <i>FEANI</i> Требования профессиональных стандартов Ассоциации предприятий компьютерных и информационных технологий. Требования работодателей: Контек, ОАО «Газпром переработка», ИХН СО РАН.
<b>P5</b>	<i>Демонстрировать</i> знание о формах организации образовательной и научной деятельности в высших учебных заведениях, <i>иметь навыки</i> преподавательской работы.	Требования ФГОС (ОК-1, 10, 16, ПК-1, 14, 15), Критерий 5 АИОР (п. 5.2.1), согласованный с требованиями международных стандартов <i>EUR-ACE</i> и <i>FEANI</i>
<b>P6</b>	Способность осуществлять организационно-управленческую и социально-ориентированную деятельность с соблюдением профессиональной этики	Требования ФГОС (ОК-5,13,16, ПК-11-13,16) Критерий 5 АИОР (п.5.2.12-13) согласованный с требованиями международных стандартов <i>EUR-ACE</i> и <i>FEANI</i>
<i>Универсальные компетенции</i>		
<b>P7</b>	<i>Активно</i> владеть <i>иностранном языком</i> на уровне, позволяющем работать в интернациональной среде, включая разработку документации и представление результатов инновационной деятельности. Толерантность в восприятии социальных и культурных различий.	Требования ФГОС (ОК-2, 3,4, 7, ПК-8). Критерий 5 АИОР (п. 5.2.11), согласованный с требованиями международных стандартов <i>EUR-ACE</i> и <i>FEANI</i> и Требования профессиональных стандартов Ассоциации предприятий компьютерных и информационных технологий

<p><b>Р8</b></p>	<p>Эффективно работать индивидуально, в качестве члена и руководителя группы, состоящей из специалистов различных направлений и квалификаций, демонстрировать ответственность за результаты работы и готовность <i>следовать корпоративной культуре</i> организации</p>	<p>Требования ФГОС (ОК-1,4, 6, ПК-8,11,12), Критерий 5 АИОР (пп. 5.2.9,5.2.13), согласованный с требованиями международных стандартов <i>EUR-ACE</i> и <i>FEANI</i> и Требования профессиональных стандартов Ассоциации предприятий компьютерных и информационных технологий. Требования работодателей: Контек, ОАО «Газпром переработка», ООО Нижневартонскэнергонефть».</p>
<p><b>Р9</b></p>	<p><i>Самостоятельно учиться и непрерывно повышать квалификацию</i> в течение всего периода профессиональной деятельности. Способность к интеллектуальному, культурному, нравственному и профессиональному саморазвитию.</p>	<p>Требования ФГОС (ОК-8,9,16, ПК-5, 11), Критерий 5 АИОР (5.2.14), согласованный с требованиями международных стандартов <i>EUR-ACE</i> и <i>FEANI</i>. Требования работодателей: Контек, ОАО «Газпром переработка», ООО Нижневартонскэнергонефть».</p>

**Министерство образования и науки Российской Федерации**  
федеральное государственное автономное образовательное учреждение  
высшего образования  
**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

---

Инженерная школа информационных технологий и робототехники  
Направление подготовки: Прикладная математика и информатика  
Отделение информационных технологий

УТВЕРЖДАЮ:  
Зав. кафедрой

\_\_\_\_\_  
(Подпись)      (Дата)      (Ф.И.О.)

**ЗАДАНИЕ**  
на выполнение выпускной квалификационной работы

В форме:

бакалаврской работы

Студенту:

Группа	ФИО
8Б51	Кулигин Сергей Михайлович

Тема работы:

Решение задачи квадратичного программирования на основе преобразования Гаусса-Жордана

Утверждена приказом директора (дата, номер)

Срок сдачи студентом выполненной работы:

**ТЕХНИЧЕСКОЕ ЗАДАНИЕ:**

<b>Исходные данные к работе</b> <i>(наименование объекта исследования; документы конференции и отчеты НИИР; программное обеспечение).</i>	Матрицы и векторы коэффициентов квадратичной и линейной части целевого функционала и ограничений.
<b>Перечень подлежащих исследованию, проектированию и разработке вопросов</b> <i>(аналитический обзор по литературным источникам с целью выяснения достижений мировой науки техники в рассматриваемой области; постановка задачи исследования, проектирования, конструирования; содержание процедуры исследования, проектирования, конструирования; обсуждение результатов выполненной работы; наименование дополнительных</i>	1) Аналитический разбор литературы по теме “квадратичное программирование”. 2) Постановка задачи. 3) Изучение вопросов и тем, необходимых для понимания и использования алгоритма. 4) Изучение и обоснование алгоритма для решения задач квадратичного программирования

<i>разделов, подлежащих разработке; заключение по работе).</i>	5) Использование и доработка алгоритма решения задач квадратичного программирования, составление программы. 6) Тестирование работоспособности алгоритма и программы. 7) Применение программы для тестового примера. 8) Заключение к работе
--	---

<b>Перечень графического материала</b> <i>(с точным указанием обязательных чертежей)</i>	
---	--

<b>Консультанты по разделам выпускной квалификационной работы</b> <i>(с указанием разделов)</i>	
--	--

Раздел	Консультант
1. Финансовый менеджмент, ресурсоэффективность и ресурсосбережение	Подопригора Игнат Валерьевич
2. Социальная ответственность	Мезенцева Ирина Леонидовна

<b>Дата выдачи задания на выполнение выпускной квалификационной работы по линейному графику</b>	
---	--

**Задание выдал руководитель:**

Должность	ФИО	Ученая степень, звание	Подпись	Дата
доцент ОМИ ШБИП	Вылегжанин Олег Николаевич	к.х.н, доцент		

**Задание принял к исполнению студент:**

Группа	ФИО	Подпись	Дата
8Б51	Кулигин Сергей Михайлович		

**Министерство образования и науки Российской Федерации**  
Федеральное государственное автономное образовательное учреждение  
высшего образования  
**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

---

Институт кибернетики

Направление подготовки Прикладная математика и информатика

Уровень образования бакалавр

Кафедра программной инженерии

Период выполнения \_\_\_\_\_ осенний / весенний семестр 2018/2019 учебного года

Форма представления работы:

бакалаврская работа

**КАЛЕНДАРНЫЙ РЕЙТИНГ-ПЛАН**  
**выполнения выпускной квалификационной работы**

Срок сдачи студентом выполненной работы:

Дата контроля	Название раздела (модуля) / вид работы (исследования)	Максимальный балл раздела (модуля)
08.02.2019	Постановка целей и задач, получение исходных данных	
16.02.2019	Подбор и изучение материалов по тематике	
27.03.2019	Разработка календарного плана	
01.03.2019	Обсуждение литературы	
11.03.2019	Выбор метода для реализации	
27.03.2019	Изучение и доработка алгоритма решения задачи	
10.04.2019	Реализация алгоритмов в виде программы	
30.04.2019	Проведение расчетов	
14.05.2019	Оформление расчётно-пояснительной записки	
21.05.2019	Оформление графического материала	
28.05.2019	Подведение итогов	

Составил преподаватель:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
доцент ОМИ ШБИП	Вылегжанин Олег Николаевич	к.х.н, доцент		

**СОГЛАСОВАНО:**

Зав. кафедрой	ФИО	Ученая степень, звание	Подпись	Дата

## РЕФЕРАТ

Пояснительная записка выполнена на 82-х листах машинописного текста, содержит 12 рисунков, 12 таблиц, 22 источника, 7 приложений.

Ключевые слова: квадратичное программирование, учёт ограничений-равенств, система линейных ограничений, многогранник решений, алгоритм.

Объектом исследования является целевая функция, состоящая из линейной и квадратичной частей; системы линейных ограничений-равенств и ограничений-неравенств.

Целью работы является изучение и доработка алгоритма нахождения экстремума целевой функции для заданных систем линейных ограничений, построение программы на его основе.

В процессе работы были изучены алгоритм учёта ограничений-равенств и алгоритм Хилдрета-Д'Эзопо, на их основе была составлена программа. Алгоритм и программа были проверены на тестовом примере, подтверждающей их корректность.

Область применения: автоматические системы управления, оценка и обработка сигналов, коммуникации и сети, схемотехника, анализ данных и моделирование, финансы, статистика.

Экономическая эффективность работы заключается в сокращении времени, необходимого для решения задачи подобного типа, путем использования программы.



## ОПРЕДЕЛЕНИЯ

В данной работе используются следующие термины с соответствующими определениями:

**выпуклая функция:** функция, для которой любой отрезок между двумя любыми точками графика функции в векторном пространстве лежит не ниже соответствующей дуги графика;

**оператор-проектор:** матрица, построенная для заданного линейного пространства, результатом умножения произвольного вектора на которую является ортогональная пространству компонента этого вектора;

**многогранник решений:** область допустимых значений для системы линейных неравенств;

**гиперплоскость:** пространство, имеющее размерность на единицу меньшую, чем исходное пространство;

**корректность:** свойство алгоритма, означающее, что если алгоритм создан для решения определенной задачи, то для всех исходных данных он должен всегда давать правильный результат и ни для каких исходных данных не будет получен неправильный результат.

## Оглавление

Введение.....	13
Обзор литературы.....	14
1 Объект и методы исследования .....	15
1.1 Математическая постановка задачи .....	15
1.2 Анализ геометрии задачи .....	18
1.2.1 Неограниченный экстремум .....	18
1.2.2 Геометрический смысл решения .....	18
1.2.3 Роль ограничений.....	20
1.3 Разработка модели алгоритма учета ограничений-равенств.....	22
1.4. Итерационный процесс поиска конечного решения .....	25
1.5 Программная реализация .....	28
1.5.1 Выбор программной среды .....	28
1.5.2 Выбор языка разработки.....	28
2 Расчёты и аналитика .....	30
2.1 Основные операции, на которых базируются алгоритмы .....	30
2.2 Тестовый пример.....	31
3 Финансовый менеджмент, ресурсоэффективность и ресурсосбережение... 36	
3.1 Оценка коммерческого и инновационного потенциала научных исследований .....	36
3.1.1 Потенциальные потребители результатов исследований .....	36
3.1.2 Анализ конкурентных технических решений .....	36
3.2 SWOT-анализ.....	37
3.3 Планирование научно-исследовательских работ.....	38
3.3.1 Структура работ в рамках научного исследования .....	38
3.3.2 Определение трудоемкости выполнения работ .....	39
3.3.3 Разработка графика проведения научного исследования .....	40
3.4 Бюджет научно-технического исследования .....	42
3.4.1 Расчет затрат на специальное оборудование для научных работ .....	42

3.4.2 Основная заработная плата исполнителей темы.....	43
3.4.3 Дополнительная заработная плата .....	44
3.4.4 Отчисления во внебюджетные фонды .....	45
3.4.5 Прочие прямые затраты.....	46
3.4.6 Накладные расходы.....	46
3.4.7 Формирование бюджета затрат научно-исследовательского проекта ...	46
3.5 Определение ресурсной, финансовой, бюджетной, социальной и экономической эффективности исследования .....	47
4 Социальная ответственность .....	53
4.1 Правовые и организационные вопросы обеспечения безопасности .....	53
4.1.1 Специальные правовые нормы трудового законодательства.....	53
4.1.2 Организационные мероприятия при компоновке рабочей зоны .....	55
4.2 Профессиональная социальная безопасность .....	57
4.2.1 Анализ вредных и опасных факторов .....	58
4.2.2 Обоснование мероприятий по защите исследователя от действия опасных и вредных факторов .....	61
4.3 Экологическая безопасность.....	62
4.4 Безопасность в чрезвычайных ситуациях.....	63
4.4.1 Анализ вероятных ЧС.....	63
4.4.2 Обоснование мероприятий по предотвращению ЧС и разработка порядка действия в случае возникновения ЧС .....	64
Список использованных источников .....	68
Приложение А Листинг процедуры расчёта обратной матрицы .....	71
Приложение Б Листинг процедуры расчёта псевдообратной матрицы .....	72
Приложение В Листинг процедуры умножения матриц.....	73
Приложение Г Листинг процедуры поиска столбца матрицы с максимальной нормой .....	74
Приложение Д Листинг процедуры вывода матрицы.....	75
Приложение Е Листинг процедуры удаления столбца из одной матрицы и запись в другую .....	76



## Введение

Одним из основных математических аппаратов теории исследования операций является математическое программирование, которое включает в себя в виде разделов линейное, выпуклое и нелинейное программирование. Простейшими задачами, относящимися ко второму из названных разделов, являются задачи так называемого квадратичного программирования. Они характеризуются тем, что оптимизируемая функция представляет квадратичную форму, обладающую положительной или отрицательной полуопределённостью, а множество возможных решений задаётся линейными условиями [1].

В литературе отсутствует описание единого метода нахождения экстремума целевой функции, однако сама задача нахождения может быть разбита на несколько подзадач, для решения некоторых из них существуют эффективные алгоритмы.

В связи с этим возникает задача построения алгоритма, позволяющего найти экстремум целевой функции при любых заданных ограничениях.

В ходе построения алгоритма необходимо решить следующие вопросы:

- реализовать и исследовать алгоритм учета ограничений-равенств;
- разработать и исследовать алгоритм пересчёта целевой функции с учетом ограничений-равенств;
- разработать и исследовать алгоритм поиска экстремума на основе преобразования Гаусса-Жордана;
- исследовать полученный алгоритм на результативность, массовость и правильность с использованием различных примеров.

В данной работе объектом исследования является алгоритм нахождения экстремума целевой функции, а предметом – его характеристики, такие как результативность, массовость и правильность.

## Обзор литературы

Задача решения квадратичного программирования имеет множество приложений в различных проблемных областях, несмотря на это, она имеет слабое освещение в литературе.

Основополагающей книгой по данной тематике является монография [1]. Излагаемая в [1] теория охватывает все основные аспекты квадратичного программирования. В книге [1] собрано большинство методов, созданных к моменту ее написания, для каждого из которых указаны условия применения.

В [2] дано полное изложение основ теории линейного программирования, которое является краеугольным камнем теории квадратичного программирования.

В статьях [3], [4] приведено и обосновано использование оператора-проектора для задачи линейного программирования, а также процедура учета системы ограничений равенств для понижения размерности системы линейных неравенств.

В [5] дано краткое и простое изложение теории оптимизации, а также некоторые ее приложения.

В статье [11] описана часть метод решения систем линейных неравенств, а именно получение одной из вершин многогранника решений посредством итерационной процедуры с применением оператора-проектора.

## 1 Объект и методы исследования

### 1.1 Математическая постановка задачи

Пусть  $I(x)$  – выпуклая квадратичная функция

$$I(x) = x^T Qx + p^T x. \quad (1.1)$$

Здесь  $Q$  – симметричная положительно полуопределённая матрица.

Пусть, далее,

$$f_i(x) = a_i^T x - b_i. \quad (1.2)$$

Сформулируем основную задачу квадратичного программирования с линейными условиями как задачу минимизации  $I(x)$  при ограничениях

$$a_i^T x - b_i = 0, \quad i = 1, 2, \dots, m, \quad (1.3)$$

$$x \geq 0. \quad (1.4)$$

Если  $m$  векторов строк  $a_i^T$  собрать в матрицу  $A_1$  с размерами  $m \times n$  и величины  $b_i$  – в вектор  $b_1$ , то систему ограничений (1.3) можно записать в виде

$$A_1 x - b_1 = 0.$$

Эти ограничения являются ограничениями-равенствами. Аналогично, ограничения-неравенства имеют вид

$$A_2 x - b_2 \leq 0.$$

Если в квадратичных задачах, встречающихся на практике, функция цели не минимизируется, а максимизируется или если в некоторых ограничениях вместо знака  $\leq$  стоит знак  $\geq$ , то такие задачи всегда можно привести к вышеупомянутой форме. Для этого достаточно умножить на  $-1$  функцию цели и соответствующие неравенства.

И тогда квадратичную задачу можно представить в матричной записи:

$$\min\{I(x) | A_1 x = b_1, A_2 x \leq b_2, x \geq 0\}. \quad (1.5)$$

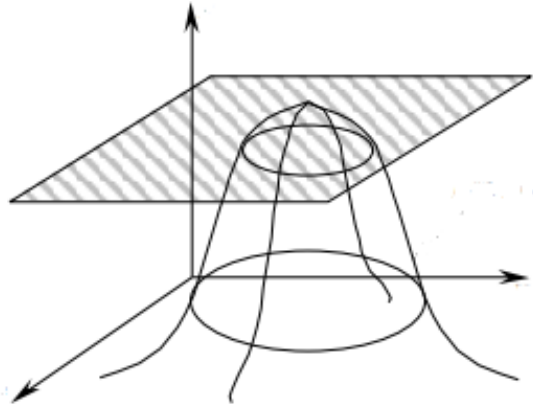


Рисунок 1.1 Графическое представление задачи

Элементы  $A$ ,  $b$  и  $p$  могут быть заданы весьма произвольно.

Рассмотрим теорему Куна-Таккера [1], занимающую центральное место в теории выпуклого программирования. Эта теорема представляет собой обобщение классического метода множителей Лагранжа для определения экстремума при наличии ограничивающих условий на случай, когда они содержат не только равенства, но и неравенства.

Существует необходимое и достаточное условие для того, чтобы некоторая точка  $\hat{x}$  представляла собой решение задачи (1.5). Критерии, которые даёт теорема, выражаются с помощью обобщённой функции Лагранжа.

Функцией Лагранжа для задачи (1.5) называется функция

$$\Phi(x, \lambda) = I(x) + \sum_{i=1}^m \lambda_i (a_{2i}^T x - b_{2i}) = x^T Qx + p^T x + \lambda^T (A_2 x - b_2), \quad (1.6)$$

от пары переменных  $x$  и  $\lambda$ . При этом координаты вектора  $\lambda$  называются множителями Лагранжа.

*Теорема Куна-Таккера.* Вектор  $\hat{x}$  представляет собой решение задачи (1.5) тогда и только тогда, когда существует вектор  $\hat{\lambda}$  такой, что

$$\hat{x} \geq 0, \hat{\lambda} \geq 0, \quad (1.7)$$

$$\Phi(\hat{x}, \lambda) \leq \Phi(\hat{x}, \hat{\lambda}) \leq \Phi(x, \hat{\lambda}) \quad (1.8)$$

для всех  $x \geq 0, \lambda \geq 0$ .

Обозначим



$$\frac{\partial \Phi}{\partial x} = v, \quad -\frac{\partial \Phi}{\partial \lambda} = y,$$

тогда

$$v = 2Qx + p + A_2^T \lambda = \frac{\partial \Phi}{\partial x}, \quad (1.9)$$

$$y = -A_2 x + b_2 = -\frac{\partial \Phi}{\partial \lambda}. \quad (1.10)$$

Условия Куна-Таккера для задачи квадратичного программирования имеют вид:

$$\begin{aligned} \text{а) } & A_2 x + y = b_2, \\ \text{б) } & 2Qx - v + A_2^T \lambda = -p, \\ \text{в) } & x^T v + y^T \lambda = 0, \\ \text{г) } & x \geq 0, v \geq 0, y \geq 0, \lambda \geq 0. \end{aligned} \quad (1.11)$$

Выполнение условий а) – г) необходимо и достаточно для существования решения квадратичной задачи. Другими словами, если существует такой  $\hat{x}$ , для которого выполняются условия Куна-Таккера, то  $\hat{x}$  – искомое решение.

## 1.2 Анализ геометрии задачи

### 1.2.1 Неограниченный экстремум

Задачу квадратичного программирования без ограничений имеет вид

$$\min\{I(x)\} = \min\{x^T Qx + p^T x\}, \quad (1.12)$$

где  $I(x)$  – выпуклая дифференцируемая функция.

Тогда, чтобы элемент  $x^* \in R^n$ , являлся решением (1.12), необходимо и достаточно чтобы градиент минимизируемой функции  $I(x)$  в точке экстремума

$$\nabla I(x^*) = \frac{\partial I(x^*)}{\partial x} = Qx^* + p = 0. \quad (1.13)$$

Действительно, если  $I(x)$  выпукла, то

$$I(x) - I(x^*) \geq \left( \frac{\partial I(x^*)}{\partial x}, x - x^* \right) = 0, \text{ для } \forall x \in R^n,$$

т.е.  $I(x) \geq I(x^*)$  и  $x^* = -Q^+p$  – решение (1.12).

### 1.2.2 Геометрический смысл решения

В задаче квадратичного программирования целевая функция геометрически представляет собой некоторое пространство, определённое на множестве, которое определяется ограничениями-неравенствами.

Каждое ограничение представляет собой гиперплоскость, которая разбивает исходное пространство на два подпространства, одно из которых удовлетворяет этому ограничению. Система ограничений в таком случае представляется выпуклым многогранником допустимых значений – общей частью  $n$ -мерного пространства, удовлетворяющей всем уравнениям этой системы (рисунок 1.2). Стрелками показано, какие области соответствуют условиям ограничений, а заштрихованная область, соответственно, является областью допустимых значений.

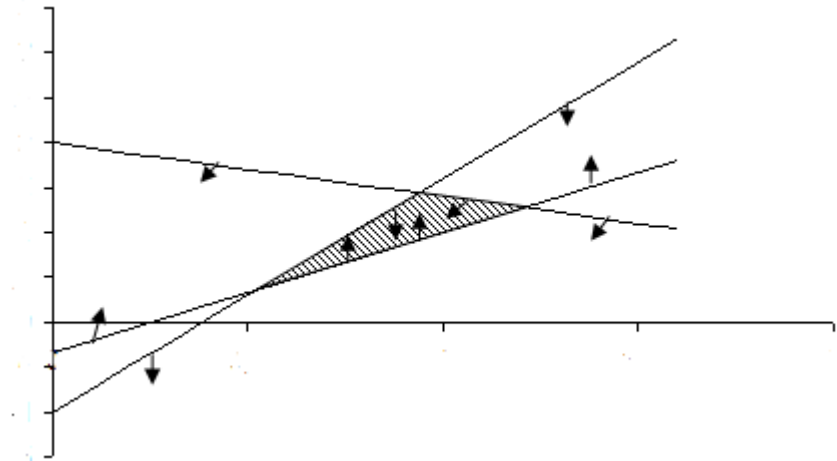


Рисунок 1.2 Многоугольник ограничений для  $n = 2$

Рассмотрим геометрическую интерпретацию квадратичной задачи. Пусть матрица  $Q$  положительно определена. В этом случае  $I(x)$  строго выпукла, линии уровня  $I(x) = const$  образуют эллипсы, общий центр которых соответствует безусловному минимуму  $I(x)$ . Нахождение решения сводится к определению той точки многогранника ограничений, которая лежит на линии уровня с наименьшим значением  $I(x)$ .

При рассмотрении двумерного случая возникает одна из трёх возможных ситуаций:

- искомое решение находится на пересечении двух рёбер многоугольника ограничений (рисунок 1.3, а);
- искомое решение находится на одном из рёбер многоугольника ограничений (рисунок 1.3, б);
- искомое решение находится внутри многоугольника допустимых значений (рисунок 1.3, в).

Похожие ситуации возникают и в случае  $n > 2$ .

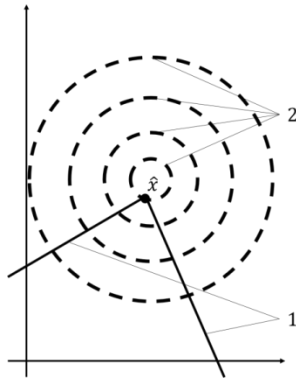


Рисунок 1.3, а

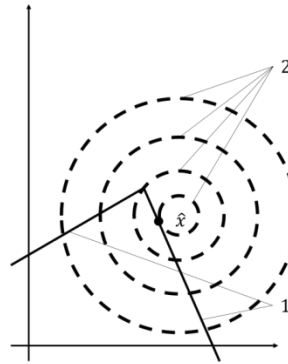


Рисунок 1.3, б

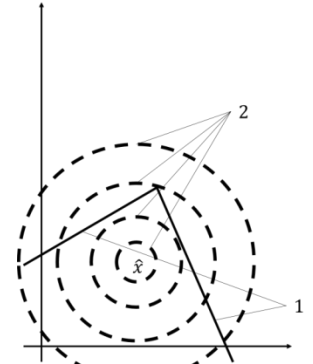


Рисунок 1.3, в

где 1 – рёбра многоугольника ограничений, 2 – линии уровня целевой функции  $I(x)$ ,  $\hat{x}$  – искомое решение задачи КП.

### 1.2.3 Роль ограничений

Во время решения практически любой реальной задачи математического программирования возникает ряд ограничений, наложенных на целевой функционал и связанных с различными факторами, например, конечный запас ресурсов. Такие ограничения определяются в виде системы, состоящей из уравнений и неравенств вида

$$a_{i1}^1 x_1 + \dots + a_{in}^1 x_n = b_i^1 \quad (1.14)$$

$$a_{i1}^2 x_1 + \dots + a_{in}^2 x_n \leq b_i^2 \quad (1.15)$$

$$x_1 \geq 0, \dots, x_n \geq 0, i = \overline{1, m}, \quad (1.16)$$

где  $a_{in}^1$  – элементы матрицы ограничений-равенств  $A_1$ ,  $a_{in}^2$  – элементы матрицы ограничений-неравенств  $A_2$ .

Каждое неравенство такой системы геометрически определяет плоскость  $a_{i1}^2 x_1 + \dots + a_{in}^2 x_n = b_i^2, i = \overline{1, m}$ . Условия неотрицательности (1.16) определяют полупространства в соответствии с граничными прямыми  $x_1 = 0, \dots, x_n = 0$ . Пересечение таких выпуклых плоскостей образует их общую часть пространства, которая также является выпуклым множеством и представляет собой совокупность точек, называемую областью допустимых

значений [2]. Координаты каждой точки составляют решение системы (1.14)-(1.16).

Для системы, где  $n = 3$ , ограничения-неравенства определяют подпространство исходного пространства с граничными плоскостями  $a_{i1}^2 x_1 + a_{i2}^2 x_2 + a_{i3}^2 x_3 = b_i^2$ , а условия неотрицательности – полупространства с граничными плоскостями  $x_1 = 0, x_2 = 0, x_3 = 0$ . Пересекаясь, эти множества образуют один из трёх возможных видов области допустимых значений:

- а) многогранник;
- б) конус;
- в) пересечение многогранника и конуса.

Таким образом, геометрический смысл ограничений-неравенств задачи квадратичного программирования состоит в отыскании такой области допустимых значений, в которой любая точка удовлетворяет всем ограничениям одновременно.

Рассмотрим геометрическую интерпретацию ограничений-равенств. Каждое из таких ограничений представляет собой подпространство размерности, на единицу меньше, чем исходное пространство, то есть  $n - 1$ , а пересечение двух таких подпространств даёт ещё одно подпространство размерности меньшей ещё на единицу, то есть  $n - 2$ , причём в нём выполняются оба ограничения. Таким образом, роль ограничений-равенств состоит в уменьшении размерности пространства, которому принадлежит определяемый вектор. Пространство уменьшается на величину ранга матрицы ограничений-равенств.

### 1.3 Разработка модели алгоритма учета ограничений-равенств

Учёт ограничений-равенств осуществляется при помощи процесса рекуррентного псевдообращения, описанного в [3]. Описывается алгоритм, позволяющий сократить размерность пространства искомого вектора неизвестных при решении оптимизационных задач с ограничениями-равенствами. Схема алгоритма представлена на рисунке 1.4.

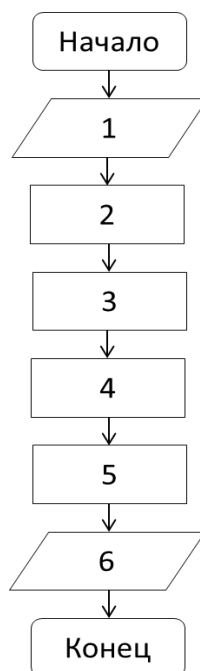


Рисунок 1.4 Схема общего алгоритма учёта ограничений-равенств

1. Ввод начальных данных: матрица коэффициентов квадратичной части функционала  $Q$ , вектор коэффициентов линейной части функционала  $p$ , матрица ограничений-равенств  $A_1$ , матрица ограничений-неравенств  $A_2$ , вектор свободных элементов ограничений-равенств  $b_1$ , точность  $tol$ .

2. Расчёт псевдообратной матрицы  $A_1^+$  для матрицы ограничений-равенств  $A_1$ . Этот пункт разбивается на несколько подпунктов:

2.1. Поиск столбца с максимальной нормой матрицы  $A_1$  с помощью формулы

$$\|A_{1_i}\| = \sqrt{A_{1_i}^T \times A_{1_i}} \quad (1.17)$$

2.2. Запись найденного столбца в новую матрицу  $An$  и удаление его из матрицы  $A_1$ . Столбец и элемент с этими номерами удаляются из  $A_2$  и  $p$  соответственно.

2.3. Расчёт псевдообратной матрицы  $An^+$  с помощью формулы

$$An^+ = \frac{An^T}{An^T \times An}. \quad (1.18)$$

3. Процесс рекуррентного псевдообращения. Этот пункт также включает в себя подпункты, схема представлена на рисунке 1.5.

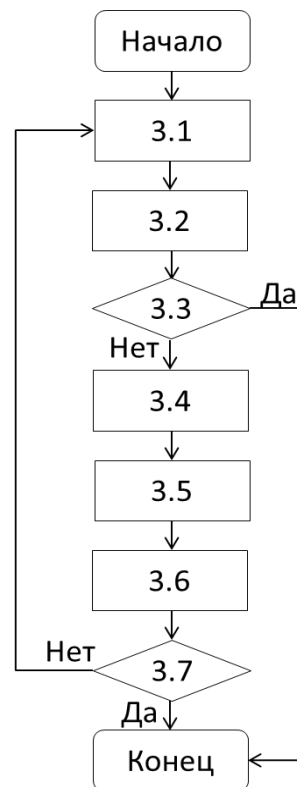


Рисунок 1.5 Схема процесса рекуррентного псевдообращения

3.1. Расчёт новой вспомогательной матрицы  $C$  по формуле

$$C = A_1 - An \times An^+ \times A_1. \quad (1.19)$$

3.2. Поиск столбца с максимальной нормой матрицы  $C$  с использованием (1.17).

3.3. Превышает ли норма найденного столбца заданной точности  $tol$ .

3.4. Удаление столбца с соответствующим номером из  $A_1$  и запись его в  $An$ . Удаление столбца с соответствующим номером из  $A_2$ . Удаление элемента с соответствующим номером из  $p$  и запись его в  $pn$ .

3.5. Расчёт псевдообратной матрицы  $C^+$  при помощи (1.18).

3.6. Пересчёт  $An^+$  на основе  $C^+$  с помощью (1.18).

3.7. Пуста ли матрица  $A_1$ .

4. Пересчёт коэффициентов целевой функции. Он включает в себя следующие подпункты:

4.1. Разбиение матрицы  $Q$  по следующему правилу:

$$Q = \begin{bmatrix} Q_0 & \hat{Q} \\ \bar{Q} & \tilde{Q} \end{bmatrix}.$$

4.2. Пересчёт матрицы  $Q$  по формуле:

$$Q' = W^T Q_0 W - \bar{Q} W + W^T \hat{Q} + \tilde{Q}, \quad (1.20)$$

где  $W = An^+ \times A_1$ .

4.3. Пересчёт вектора  $p$  по формуле:

$$p' = p^T - pn^T An^+ A_1 + b_1^T * (\tilde{Q}^T + \hat{Q} - (Q_0 + Q_0^T)W). \quad (1.21)$$

5. Пересчёт вектора свободных элементов  $b_1$  по формуле:

$$b_1' = An^+ \times b_1. \quad (1.22)$$

6. Вывод преобразованных матриц  $Q', An, A_2$  и векторов  $p'$  и  $b_1'$ .



#### 1.4. Итерационный процесс поиска конечного решения

Предложен асимптотический метод определения экстремума квадратичных форм [1], сущность которого состоит в решении двойственных задач. Данный метод называется метод Хилдрета-Д'Эзопо. Этот метод особенно удобен для вычислительных машин, отдельные шаги итерационного процесса чрезвычайно просты, поэтому случайные сбои быстро исправляются. При этом целевая функция должна быть строго выпуклой, что несколько ограничивает применимость метода.

Рассмотрим задачу:

$$\min\{I(x) | A_2 x \leq b_2, x \geq 0\}, \quad (1.23)$$

где  $A_2$  – матрица ограничений-неравенств.

Ограничения-равенства не рассматриваются, так как предполагается, что их нет или они были учтены на предыдущих этапах решения. Симметричная матрица  $Q$  предполагается положительно определённой.

В силу предположений о функции цели и ограничениях задача (1.23) имеет единственное решение  $\hat{x}$ .

Условия Куна-Таккера для задачи (1.23) имеют вид:

$$\begin{aligned} \text{а) } & A_2 x + y = b_2, \\ \text{б) } & 2Qx + A_2^T \lambda = -p, \\ \text{в) } & \lambda \geq 0, y \geq 0, \\ \text{г) } & \lambda^T y = 0. \end{aligned} \quad (1.24)$$

Так как матрица  $Q$  положительно определена, то существует обратная матрица  $Q^{-1}$  и условие б) можно разрешить относительно  $x$ :

$$x(\lambda) = -\frac{1}{2} Q^{-1} (A_2^T \lambda + p). \quad (1.25)$$

Тогда условия Куна-Таккера примут вид:

$$\begin{aligned} 2G\lambda - y &= -h, \\ \lambda \geq 0, y &\geq 0, \end{aligned} \quad (1.26)$$

$$\lambda^T y = 0,$$

где для удобства введены обозначения

$$h = \frac{1}{2} A_2 Q^{-1} p + b_2, \quad (1.27)$$

$$G = \frac{1}{4} A_2 Q^{-1} A_2^T \quad (1.28)$$

и первое уравнение умножено на  $-1$ .

Любой задаче математического программирования можно поставить в соответствие другую задачу, которая называется двойственной, а совокупность исходной и двойственной к ней задаче называется двойственной системой. Каждому неравенству или равенству в одной системе соответствует в другой некоторая неотрицательная переменная, и наоборот. Матрица коэффициентов одной системы представляет собой транспонированную матрицу коэффициентов другой системы, взятую со знаком минус.

Условия (1.26) являются условиями Куна-Таккера для двойственной задачи

$$\max\{\varphi(\lambda) = h^T \lambda + \lambda^T G \lambda \mid \lambda \geq 0\}. \quad (1.29)$$

Поскольку условия Куна-Таккера необходимы и достаточны и, будучи составленные для задач (1.23) и (1.29), фактически совпадают, то  $\hat{x}$  является решением задачи (1.23) тогда и только тогда, когда  $\hat{x} = x(\hat{\lambda})$ , причём  $\hat{\lambda}$  представляется собой решение задачи (1.29) [1].

Задача (1.29) является двойственной к (1.23). Двойственная задача разрешима, если разрешима прямая задача (1.23), и для получения решения  $\hat{x}$  прямой задачи достаточно найти решение  $\hat{\lambda}$  двойственной задачи и воспользоваться формулой (1.25).

Если решение (1.29) не единственно, то все  $\hat{\lambda}$ , естественно, дают равные значения  $\hat{x}$ .

Предложенный метод для решения (1.29) представляет собой некоторое видоизменение итерационного метода Гаусса-Жордана для решения систем линейных уравнений, который базируется на элементарных преобразованиях.

Решение начинается с произвольной допустимой точки  $\lambda^0 \geq 0$ . Удобно выбрать  $\lambda^0 = 0$ , так как в окончательном решении некоторые  $\hat{\lambda}_i$  имеют значение 0. Компоненты следующей итерационной точки  $\lambda^1$  определяются по очереди в результате минимизации  $\varphi(\lambda)$  по каждой компоненте  $\lambda_i$  вектора  $\lambda$  при ограничении  $\lambda_i \geq 0$ , причём остальные компоненты сохраняют значения, полученные в последний раз. Аналогично получаются  $\lambda^2$  из  $\lambda^1$  и так далее. Вообще,  $\lambda^{k+1}$  для  $k = 0, 1, 2, \dots$  находят по формулам:

$$\lambda_i^{k+1} = \max\{0, \omega_i^{k+1}\}, \quad (1.30)$$

$$\omega_i^{k+1} = -\frac{1}{g_{ii}} \left( \sum_{j=1}^{i-1} g_{ij} \lambda_j^{k+1} + \frac{h_i}{2} + \sum_{j=i+1}^m g_{ij} \lambda_j^k \right), \quad (1.31)$$

$$i = 1, 2, \dots, m,$$

$\omega_i^{k+1}$  – те значения  $\lambda_i$ , которые получаются, если положить производные

$$\frac{\partial \varphi}{\partial \lambda_i} = 2 \sum_{j=1}^m g_{ij} \lambda_j + h_i,$$

равными нулю, для  $j < i$  подставить вновь вычисленные значения  $\lambda_j^{k+1}$ , а для  $j > i$  взять значения  $\lambda_j^k$ .

Обозначим вектор, составленный из тех переменных  $\lambda_j$ , которые в окончательном решении положительны, через  $\lambda'$ . Условия (1.26) при этом упрощаются, поскольку  $y_j = 0$ :

$$2G'\lambda' + h' = 0, \quad (1.32)$$

причём  $G'$  получается из  $G$  вычёркиванием тех строк и столбцов, которые соответствуют  $\lambda_j = 0$ . Аналогично получается  $h'$  и  $h$ .

## **1.5 Программная реализация**

### **1.5.1 Выбор программной среды**

Для реализации разработанных алгоритмов могут быть использованы такие типы программного обеспечения, как специализированные математические пакеты (MATLAB, Maple) или различные объектно-ориентированные языки программирования (C++, Pascal, Delphi).

Преимуществами математических пакетов является запись сложных математических выражений в исходном виде, оптимизированные алгоритмы численных и аналитических расчётов, а также возможность создания встроенными средствами высококачественных таблиц, графиков и текстовых документов. Однако, большая часть программного обеспечения такого типа ограничена базовым функционалом или имеет узкую специализацию в программировании сложных задач.

Для решения таких задач, как создание готового продукта для проведения вычислений в области математического программирования, наиболее подходящим вариантом являются объектно-ориентированные языки программирования. Такие языки содержат необходимые средства для создания программ любого назначения: от низкоуровневых до сложнейших программных комплексов.

### **1.5.2 Выбор языка разработки**

Язык программирования предназначен для написания компьютерных программ, которые представляют собой набор правил, позволяющих компьютеру выполнить тот или иной вычислительный процесс, организовать управление различными объектами, и т. п.

Для реализации всех необходимых для решения задачи квадратичного программирования был выбран язык программирования C++. Это компилируемый язык программирования общего назначения, который имеет ряд преимуществ перед другими языками программирования.

В свою очередь, C++ был выбран из-за возможности разработки программ для самых различных платформ и систем и создания обобщённых алгоритмов для разных типов данных, их специализации, и вычисления на этапе компиляции. Эти возможности позволят создать программу, которая не будет иметь ограничения для реализации решения задачи квадратичного программирования.

## 2 Расчёты и аналитика

### 2.1 Основные операции, на которых базируются алгоритмы

В рассмотренных алгоритмах присутствует несколько математических операций над матрицами, применяемых для также и для реализации этих алгоритмов в программе. Операции над матрицами не реализованы отдельно в среде Visual Studio C++, поэтому необходимо программировать их самостоятельно, чтобы, в итоге, можно было считать алгоритм решения задачи квадратичного программирования реализованным полностью.

Рассмотрим подробно эти операции.

Листинг операции поиска обратной матрицы представлен в приложении А. Данный процесс представляет собой вычисление обратной матрицы. Матрица должна быть положительно определённой.

Также реализован поиск псевдообратной матрицы. Листинг представлен в приложении Б. Вычисление проводится с помощью (1.18).

Умножение матриц является самой часто используемой операцией. Она многократно используется не только в теле программы, но и почти во всех внешних функциях. Реализованный алгоритм умножения матриц также подходит и для умножения матрицы на вектор и умножения вектора на матрицу. Листинг представлен в приложении В.

Для упрощения реализации поиска задачи квадратичного решения была введена функция, которая позволяла быстро найти максимальную норму столбца матрицы. Максимальная норма столбца рассчитывается по формуле (1.17). Листинг представлен в приложении Г.

В реализуемом алгоритме присутствует операция перестановки столбцов матриц. Это значит, что необходимо удалить определённый столбец из одной матрицы и записать его в другую. Листинг данной процедуры представлен в приложении Е.

Также, для того, чтобы сократить код программы был отдельно реализован вывод матрицы на экран (приложение Д). Данная процедура подходит и для векторов.

## 2.2 Тестовый пример

Тестовый пример был использован для проверки корректности использованного алгоритма и составленной согласно этому алгоритму программы.

Исходный функционал:

$$I(x) = 1x_1 - 2x_2 + x_3 - 3.424x_4 + 0.5x_5 + 1.746x_1^2 - 0.043x_2^2 - 1.798x_3^2 - 0.402x_4^2 - 0.272x_5^2 + 1.277x_1x_2 + 1.556x_1x_3 - 0.129x_1x_4 + 0.308x_1x_5 + 1.561x_2x_3 + 0.063x_2x_4 + 0.459x_2x_5 - 0.689x_3x_4 + 0.387x_3x_5 + 0.334x_4x_5.$$

Исходная система линейных ограничений-равенств:

$$\begin{aligned} -2.901x_1 - 2.157x_2 + 0.206x_3 - 0.616x_4 - 1.198x_5 &= -0.5; \\ 0.099x_1 + 0.771x_2 + 0.306x_3 + 0.011x_4 - 0.764x_5 &= 0.5; \\ -0.409x_1 - 0.674x_2 - 0.132x_3 + 1.018x_4 + 0.181x_5 &= 4.5; \\ 0.513x_1 + 0.74x_2 - 0.016x_3 + 0.279x_4 + 0.162x_5 &= 0. \end{aligned}$$

Исходная система линейных ограничений-неравенств:

$$\begin{aligned} 0.4x_1 - 2x_2 + 0.1x_3 + 0.501x_4 + 0.35x_5 &\leq 2.225; \\ -1x_1 + 0.1x_2 - 2x_3 + 9.994x_4 - 3.575x_5 &\leq 11.037; \\ -0.5x_1 + 0.1x_2 + 0.5x_3 - 7.481x_4 + 5.425x_5 &\leq 6.537; \\ 4.5x_1 - 3x_2 - 10.1x_3 - 16.551x_4 - 30.05x_5 &\leq 110.225; \\ 4.5x_1 - 3x_2 - 2.398x_4 - 5x_5 &\leq -12; \\ -1x_1 + 2x_2 + 0.5x_3 - 1.2x_4 - 0.35x_5 &\leq -1.125; \\ 2.4x_1 - 1x_2 - 3.5x_3 + 4x_4 - 5.2x_5 &\leq 0.2; \\ -1x_1 + 1x_2 - 3x_3 + 4.687x_4 - 1.25x_5 &\leq -2.625. \end{aligned}$$

Исходная погрешность:  $tol = 0.001$ .

Матрицы функционала и систем ограничений имеют вид:

$$Q = \begin{pmatrix} 1.746 & 1.277 & 1.556 & -0.129 & 0.308 \\ 1.277 & -0.043 & 1.561 & 0.063 & 0.459 \\ 1.556 & 1.561 & -1.798 & -0.689 & 0.387 \\ -0.129 & 0.063 & -0.689 & -0.402 & 0.334 \\ 0.308 & 0.459 & 0.387 & 0.334 & -0.272 \end{pmatrix}, p = \begin{pmatrix} 1 \\ -2 \\ 1 \\ -3.424 \\ 0.5 \end{pmatrix}.$$

$$A_1 = \begin{pmatrix} -2.901 & -2.157 & 0.206 & -0.616 & -1.198 \\ 0.099 & 0.771 & 0.306 & 0.011 & -0.764 \\ -0.409 & -0.674 & -0.132 & 1.018 & 0.181 \\ 0.513 & 0.74 & -0.016 & 0.279 & 0.162 \end{pmatrix}, b_1 = \begin{pmatrix} -0.5 \\ 0.5 \\ 4.5 \\ 0 \end{pmatrix}.$$

$$A_2 = \begin{pmatrix} 0.4 & -2 & 0.1 & 0.501 & 0.35 \\ -1 & 0.1 & -2 & 9.994 & -3.575 \\ -0.5 & 0.1 & 0.5 & -7.481 & 5.425 \\ 4.5 & -3 & -10.1 & -16.551 & -30.05 \\ 4.5 & -3 & 0 & -2.398 & -5 \\ -1 & 2 & 0.5 & -1.2 & -0.35 \\ 2.4 & -1 & -3.5 & 4 & -5.2 \\ -1 & 1 & -3 & 4.687 & -1.25 \end{pmatrix}, b_2 = \begin{pmatrix} 2.225 \\ 11.037 \\ 6.537 \\ 110.225 \\ -12 \\ -1.125 \\ 0.2 \\ -2.625 \end{pmatrix}.$$

Применим рассмотренный в главе 1 алгоритм учёта ограничений-равенств. Матрица коэффициентов  $Q$  и вектор коэффициентов  $p$  после учёта ограничений-равенств:

$$Q = \begin{pmatrix} 3.84661 & 0.602051 \\ 0.602051 & -1.4698 \end{pmatrix}, p = \begin{pmatrix} -3.14139 \\ -2.18055 \end{pmatrix}$$

Система ограничений-неравенств после учета ограничений-равенств:

$$A_2 = \begin{pmatrix} -2 & 0.1 \\ 0.1 & -2 \\ 0.1 & 0.5 \\ -3 & -10.1 \\ -3 & 0 \\ 2 & 0.5 \\ -1 & -3.5 \\ 1 & -3 \end{pmatrix}, b_2 = \begin{pmatrix} 2.225 \\ 11.037 \\ 6.537 \\ 110.225 \\ -12 \\ -1.125 \\ 0.2 \\ -2.625 \end{pmatrix}.$$

Далее применим итерационный алгоритм поиска решения задачи квадратичного программирования, который также был описан в главе 1.

Сначала вычисляем обратную матрицу

$$Q^{-1} = \begin{pmatrix} 0.244 & 0.1 \\ 0.1 & -0.639 \end{pmatrix}$$

и затем  $h$  и  $G$  по формулам (1.27) и (1.28) соответственно.

Тогда получим



$$h = \begin{pmatrix} 3.265 \\ 9.908 \\ 6.758 \\ 106.25 \\ -10.521 \\ -1.841 \\ -1.197 \\ 0.512 \end{pmatrix},$$

$$G = \begin{pmatrix} 0.233 & 0.12 & -0.045 & 1.026 & 0.359 & -0.272 & 0.351 & 0.078 \\ 0.12 & -0.649 & 0.157 & -3.122 & 0.132 & 0.073 & -1.084 & -1.01 \\ -0.045 & 0.157 & -0.037 & 0.726 & -0.056 & -1.477 * 10^{-3} & 0.252 & 0.251 \\ 1.026 & -3.122 & 0.726 & -14.24 & 1.308 & -0.102 & -4.952 & -5.054 \\ 0.359 & 0.132 & -0.056 & 1.308 & 0.55 & -0.404 & 0.446 & 0.042 \\ -0.272 & 0.073 & -1.477 * 10^{-3} & -0.102 & -0.404 & 0.254 & -0.03 & 0.224 \\ 0.351 & -1.084 & 0.252 & -4.952 & 0.446 & -0.03 & -1.722 & -1.752 \\ 0.078 & -1.01 & 0.251 & -5.054 & 0.042 & 0.224 & -1.752 & -1.528 \end{pmatrix}$$

Вычислив  $G$  и  $h$ , имеем, по существу, таблицу коэффициентов для вычисления итерационных точек  $\lambda^p = \|\lambda_1^p, \lambda_2^p, \lambda_3^p, \lambda_4^p, \lambda_5^p, \lambda_6^p, \lambda_7^p, \lambda_8^p\|$ . Для этого воспользуемся формулами (1.30) и (1.31).

Начиная с

$$\lambda_1^0 = 0, \lambda_2^0 = 0, \lambda_3^0 = 0, \lambda_4^0 = 0, \lambda_5^0 = 0, \lambda_6^0 = 0, \lambda_7^0 = 0, \lambda_8^0 = 0,$$

мы получим последовательность итерационных точек:

$$1 \text{ итерация: } \lambda_1^1 = 0, \lambda_2^1 = 7.63591, \lambda_3^1 = 118.834, \lambda_4^1 = 8.15495, \\ \lambda_5^1 = 0.4435, \lambda_6^1 = 3.61802, \lambda_7^1 = 0, \lambda_8^1 = 0,$$

$$2 \text{ итерация: } \lambda_1^2 = 0, \lambda_2^2 = 0, \lambda_3^2 = 173.111, \lambda_4^2 = 12.5089, \\ \lambda_5^2 = 0, \lambda_6^2 = 7.54913, \lambda_7^2 = 0, \lambda_8^2 = 0.$$

По формуле (1.32), где

$$\lambda' = \begin{pmatrix} \lambda_3 \\ \lambda_4 \\ \lambda_6 \end{pmatrix}, G' = \begin{pmatrix} -0.037 & 0.726 & -1.477 * 10^{-3} \\ 0.726 & -14.24 & -0.102 \\ -1.477 * 10^{-3} & -0.102 & 0.254 \end{pmatrix}, \\ h' = \begin{pmatrix} 6.758 \\ 106.25 \\ -1.841 \end{pmatrix},$$

вычислим  $\lambda'$ . Тогда получим  $\lambda_3 = 37540, \lambda_4 = 1910, \lambda_5 = 989.044$ .

Все полученные значения положительны, значит, нет необходимости продолжать итерации, так как условия (1.26) выполнены. Тогда решение двойственной задачи:

$$\hat{\lambda} = \|0, 0, 173.111, 12.5089, 0, 7.54913, 0, 0\|.$$

Решение прямой программы получается по формуле (1.25), т.е.

$$\hat{x}_1 = -4.582, \hat{x}_2 = 18.728.$$

Необходимо убедиться в правильности решения. Для этого построим область допустимых значений на графике.

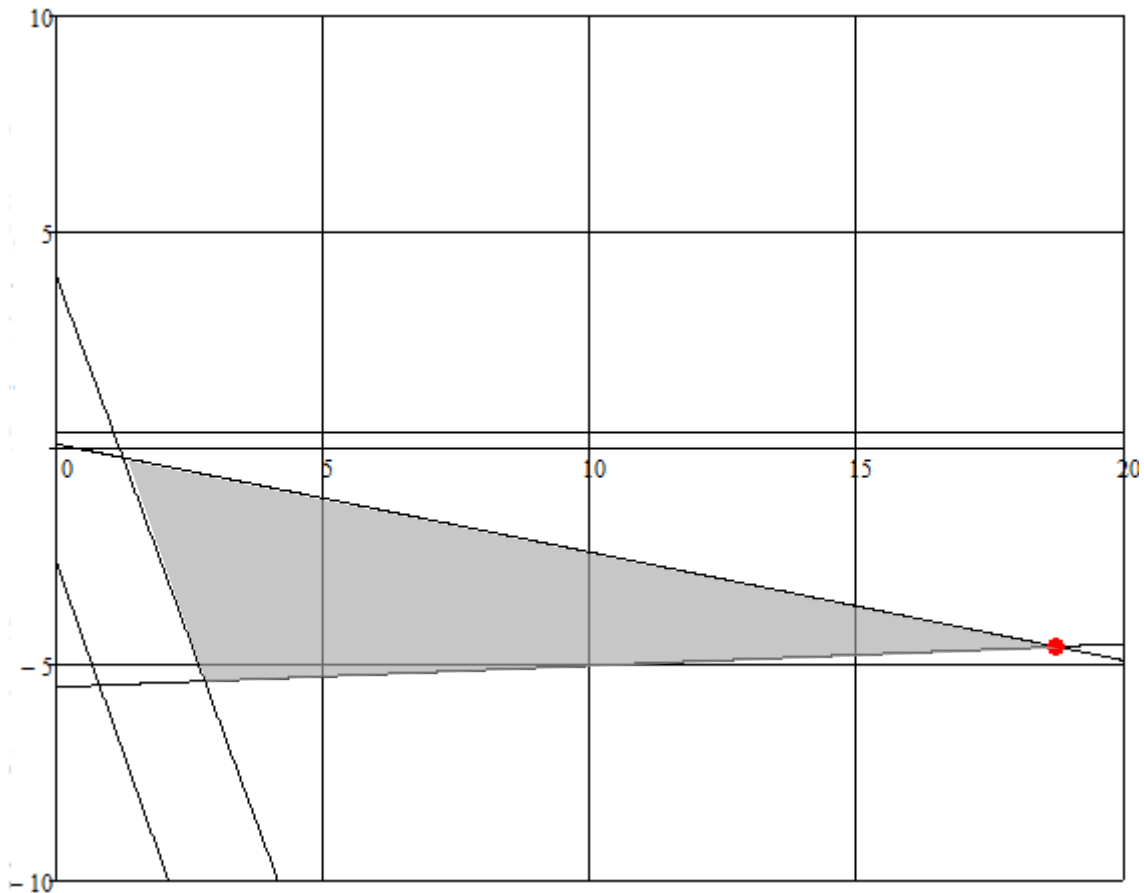


Рисунок 2.1 Область допустимых значений после учёта ограничений-равенств.

Действительно, в точке  $(-4.582, 18.728)$  значение функционала  $I(x)$  минимально и равно  $-239.626$ .

Для обоснования корректности, программа была проверена на тестовом примере, результат расчета которого вручную приведен выше, согласно использованному алгоритму. Как можно убедиться из графического решения, представленного на рисунке 2.1, решение задачи квадратичного программирования было найдено верно. Как алгоритм, так и программа дали совпадающие между собой правильные решения задачи. Значит, как алгоритм, так и программа были составлены и реализованы правильно.

**ЗАДАНИЕ ДЛЯ РАЗДЕЛА  
«ФИНАНСОВЫЙ МЕНЕДЖМЕНТ, РЕСУРСОЭФФЕКТИВНОСТЬ И  
РЕСУРСОСБЕРЕЖЕНИЕ»**

Студенту:

<b>Группа</b>	<b>ФИО</b>
8Б51	Кулигин Сергей Михайлович

<b>Школа</b>	<b>ИШИТР</b>	<b>Отделение (НОЦ)</b>	<b>Информационных технологий</b>
Уровень образования	Бакалавриат	Направление/специальность	Прикладная математика и информатика

**Исходные данные к разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»:**

1. <i>Стоимость ресурсов научного исследования (НИ): материально-технических, энергетических, финансовых, информационных и человеческих</i>	<i>Человеческие ресурсы – 2 человека (руководитель и студент-дипломник).</i>
---	--

**Перечень вопросов, подлежащих исследованию, проектированию и разработке:**

1. <i>Планирование и формирование бюджета научных исследований</i>	<i>Необходимо составить календарный план НИ и рассчитать затраты НИ.</i>
2. <i>Определение ресурсной (ресурсосберегающей), финансовой, бюджетной, социальной и экономической эффективности исследования</i>	<i>Расчет интегральных показателей эффективности исследования, выбор наилучшего исполнения.</i>

**Перечень графического материала:**

1. *Оценка конкурентоспособности технических решений*
2. *Матрица SWOT*
3. *График проведения и бюджет НИ*
4. *Оценка ресурсной, финансовой и экономической эффективности НИ*

**Дата выдачи задания для раздела по линейному графику**

**Задание выдал консультант:**

<b>Должность</b>	<b>ФИО</b>	<b>Ученая степень, звание</b>	<b>Подпись</b>	<b>Дата</b>
Доцент ОСГН ШБИП	Подопригора Игнат Валерьевич	к.э.н., доцент		

**Задание принял к исполнению студент:**

<b>Группа</b>	<b>ФИО</b>	<b>Подпись</b>	<b>Дата</b>
8Б51	Кулигин Сергей Михайлович		

### **3 Финансовый менеджмент, ресурсоэффективность и ресурсосбережение**

Целью написания раздела «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение» является проектирование и создание конкурентоспособных разработок, отвечающих современным требованиям в области ресурсоэффективности и ресурсосбережения.

#### **3.1 Оценка коммерческого и инновационного потенциала научных исследований**

##### **3.1.1 Потенциальные потребители результатов исследований**

Выполненная работа направлена на разработку и реализацию алгоритма, который будет решать задачи квадратичного программирования на основе преобразования Гаусса-Жордана.

Потенциальными потребителями результатов исследования являются крупные организации, специализирующиеся в нефтегазовой отрасли, а именно газодобывающие компании, которые имеют в своём распоряжении уже эксплуатируемые УКПГ или планируют внедрение новых установок подготовки газа. Для этих компаний разрабатывается автоматизированная система управления узла осушки газа установки комплексной подготовки газа.

##### **3.1.2 Анализ конкурентных технических решений**

Анализ конкурентных технических решений проводится с помощью оценочной карты для сравнения конкурентных технических решений, приведенной в таблице 3.1. Основным методом, используемым в данной работе, является метод Хилдрета-Д'Эзопо, который используется в поиске оптимума при попадании в область допустимых значений (осн).

Конкурентные методы: метод Баранкина-Дорфмана (бд), метод с использованием операторов-проекторов (оп).

Таблица 3.1 – Оценочная карта для сравнения конкурентных технических решений

Критерии оценки	Вес	Баллы			Конкурентоспособность		
		Б <sub>осн</sub>	Б <sub>бд</sub>	Б <sub>оп</sub>	Б <sub>осн</sub>	Б <sub>бд</sub>	Б <sub>оп</sub>
Технические критерии оценки ресурсоэффективности							
Эффективность	0,3	4	4	4	1,2	1,2	1,2
Устойчивость	0,2	5	5	5	1	1	1
Временные затраты	0,2	4	4	5	0,8	0,8	1
Новизна метода	0,1	5	3	4	0,5	0,3	0,4
Простота реализации	0,1	5	3	3	0,5	0,3	0,3
Универсальность	0,1	4	4	4	0,4	0,4	0,4
<b>Итого</b>	<b>1</b>	<b>27</b>	<b>23</b>	<b>25</b>	<b>4,4</b>	<b>4,0</b>	<b>4,3</b>

По полученным результатам можно сделать вывод, что разрабатываемый алгоритм для оценки информативности является по конкурентоспособности наиболее эффективным.

### 3.2 SWOT-анализ

SWOT-анализ – это эффективный инструмент стратегического менеджмента. SWOT-анализ заключается в исследовании внешней и внутренней сред проекта. метод, которые заключается в комплексном анализе научно-исследовательского проекта. SWOT – Strengths (сильные стороны), Weaknesses (слабые стороны), Opportunities (возможности) и Threats (Угрозы)

Разработанная для алгоритма поиска информативного атрибута матрица SWOT представлена в таблице 3.2.

Таблица 3.2 – SWOT- анализ

	<b>Сильные стороны:</b>	<b>Слабые стороны:</b>
	С1. Высокая эффективность алгоритма для решения задач квадратичного программирования; С2. Гибкость алгоритма с точки	Сл1. Узкий круг целевой аудитории; Сл2. Трудоемкий процесс написания и отладки программы для решения задач

	зрения считывания исходных данных; С3. Простая эксплуатация.	квадратичного программирования.
<b>Возможности:</b> В1. Расширение функционала; В2. Написание алгоритма на других языках программирования.	Благодаря гибкости и высокой эффективности алгоритм может быть дополнен реализацией других задач, например, задач нелинейного программирования или другого типа задач выпуклого программирования.	Написание алгоритма на другом языке программирования, а также расширение функционала могут сделать алгоритм более широко применимым, увеличить целевую аудиторию.
<b>Угрозы:</b> У1. Отсутствие спроса на продукт на рынке; У2. Развитие и появление аналогов алгоритма.	Наглядные результаты использования алгоритма, а также его низкая стоимость могут увеличить спрос на него.	Узкая направленность и использование платного языка программирования могут ослабить интерес покупателей.

### 3.3 Планирование научно-исследовательских работ

#### 3.3.1 Структура работ в рамках научного исследования

Трудоемкость выпускной квалификационной работы определяется опытным путем в человеко-днях и имеет вероятностный характер, так как зависит от трудно учитываемых факторов. Планирование комплекса предполагаемых работ осуществляется в следующем порядке:

- определение структуры работ в рамках научного исследования;
- определение участников каждой работы;
- установление продолжительности работ;
- построение графика проведения научных исследований.

В качестве исполнителей в проекте представлены руководитель и инженер. Перечень этапов, работ и исполнителей приведен в таблице 3.3.

Таблица 3.3 – Основные этапы выполнения ВКР

№	Описание этапа	Исполнители	Загруженность исполнителей
1	Составление и утверждение задания	НР	НР – 100%
2	Анализ предметной области	С, НР	С – 50%, НР – 50%
3	Разработка календарного плана	С, НР	С – 10%, НР – 90%
4	Обзор литературы и интернет - источников	С, НР	С – 50%, НР – 50%
5	Обзор задач и методов решения задач ЛП	С	С – 100%
6	Математическая постановка задачи	С, НР	С – 10%, НР – 90%
7	Выбор метода решения поставленной задачи	С, НР	С – 10%, НР – 90%
8	Разработка алгоритмов решения задачи	С	С – 100%
9	Программная реализация компьютерной модели	С	С – 100%
10	Проведение исследований на модельных данных	С, НР	С – 80% НР – 20%
11	Проведение исследований на реальных данных	С, НР	С – 60% НР – 40%
12	Расчет экономической эффективности научно-технической продукции	С	С – 100%
13	Оценка социальной ответственности проекта	С	С – 100%
14	Написание пояснительной записки	С, НР	С – 90% НР – 10%

### 3.3.2 Определение трудоемкости выполнения работ

Трудовые затраты в большинстве случаев образуют основную часть стоимости разработки, поэтому важным моментом является определение трудоемкости работ каждого из участников научного исследования.

Трудоемкость выполнения научного исследования оценивается экспертным путем в человеко-днях и носит вероятностный характер, т.к. зависит от множества трудно учитываемых факторов. Для определения ожидаемого (среднего) значения трудоемкости используется следующая формула:

$$t_{ож} = \frac{3t_{min} + 2t_{max}}{5}, \quad (3.1)$$

где  $t_{min}$  – предположительно минимальная продолжительность этапа в рабочих днях, определяемая методом экспертной оценки,  $t_{max}$  – предположительно максимальная продолжительность этапа в рабочих днях, определяемая методом экспертной оценки.

Исходя из ожидаемой трудоемкости работ, определяется продолжительность каждой работы в рабочих днях  $T_p$ , учитывающая параллельность выполнения работ несколькими исполнителями.

$$T_p^i = \frac{t_{ож}^i}{\chi_i}, \quad (3.2)$$

где  $T_p^i$  – продолжительность одной работы, раб. дн.,  $t_{ож}^i$  – ожидаемая трудоемкость выполнения одной работы, чел.-дн,  $\chi_i$  – численность исполнителей, выполняющих одновременно одну и ту же работу на данном этапе, чел.

Продолжительность каждого этапа рассчитывается по формуле:

$$t_{раб} = t_{ож} * k_d, \quad (3.3)$$

где  $t_{раб}$  – длительность этапов в рабочих днях,  $k_d$  – коэффициент, учитывающий дополнительное время на консультации и согласование работ ( $k_d = 1.2$ ).

### 3.3.3 Разработка графика проведения научного исследования

В данном пункте необходимо построить диаграмму Ганта – горизонтально-ленточный график, на котором работы представляются протяженными во времени отрезками, характеризующимися датами начала и окончания выполнения приведенных работ.

Линейный график строится на основании полученных значений  $t_{раб}$ , предварительно переведенных в календарные дни по формуле:

$$t_k = t_{раб} * K_n, \quad (3.4)$$

где  $t_k$  – длительность этапов работ в календарных днях,  $K_n$  – коэффициент календарности.

Коэффициент календарности рассчитывается по формуле:



$$K_H = \frac{T_K}{T_K - T_{ВД} - T_{ПД}}, \quad (3.5)$$

где  $T_K$  – количество календарных дней ( $T_K = 365$ ),  $T_{ВД}$  – количество выходных дней ( $T_{ВД} = 108$ ),  $T_{ПД}$  – количество праздничных дней ( $T_{ПД} = 10$ ),

$$K_H = \frac{365}{365 - 108 - 10} = 1.478,$$

Все расчеты сведены в таблицу 3.4.

Таблица 3.4 – Временные показатели проведенных работ

№	Исполнители	Продолжительность работ в днях			Трудоемкость	
		$t_{min}$	$t_{max}$	$t_{ож}$	$t_{раб}$	$t_k$
1	НР	1	2	1,4	1,68	2,48
2	С, НР	2	3	2,4	2,88	4,26
3	С, НР	1	2	1,4	1,68	2,48
4	С, НР	3	5	3,8	4,56	6,74
5	С	2	5	3,2	3,84	5,68
6	С, НР	3	6	4,2	5,04	7,45
7	С, НР	5	8	6,2	7,44	11,00
8	С	3	6	4,2	5,04	7,45
9	С	4	6	4,8	5,76	8,51
10	С, НР	2	5	3,2	3,84	5,68
11	С, НР	2	4	2,8	3,36	4,97
12	С	5	7	5,8	6,96	10,29
13	С	4	5	4,4	5,28	7,80
14	С, НР	9	12	10,2	12,24	18,09
<b>Итого</b>				<b>58</b>	<b>69,6</b>	<b>102,87</b>

На основании таблицы 3.4 построим диаграмму Ганта (Рисунок 3.1).

№ работ	Вид работ	Исполнители	T <sub>кi</sub> , кал. дн.	Продолжительность выполнения работ														
				февр		март			апрель			май			июнь			
				2	3	1	2	3	1	2	3	1	2	3	1	2		
1	Выбор направления исследований	Руководитель, инженер	3, 3	█														
2	Составление и утверждение научного задания	Руководитель	5	█														
3	Подбор и изучение материалов по теме	Руководитель, инженер	11, 11		█													
4	Календарное планирование работ по теме	Инженер	5			█												
5	Проведение теоретических расчетов и обоснований	Инженер	58			█	█	█	█	█	█	█	█	█	█			
6	Анализ полученных результатов	Руководитель, инженер	8, 8										█					

▨ – инженер

█ – руководитель

Рисунок 3.1 – Диаграмма Ганта

### 3.4 Бюджет научно-технического исследования (НТИ)

#### 3.4.1 Расчет затрат на специальное оборудование для научных работ

В данную статью включают все затраты, связанные с приобретением специального оборудования, необходимого для проведения работ по конкретной теме.

В ходе выполнения НТИ использовалось имеющееся компьютерное оборудование, поэтому его стоимость учитывается в калькуляции в виде амортизационных отчислений за 5 месяцев (таблица 3.5).

Таблица 3.5 – Расчет бюджета затрат на приобретение спецоборудования для научных работ

Наименование	Единица измерения	Количество	Цена за единицу, руб.	Затраты на материалы, руб.
Амортизация	шт.	1	16000	16000

оборудования				
<b>Итого</b>				16

### 3.4.2 Основная заработная плата исполнителей темы

Основная заработная плата руководителя рассчитывается на основании отраслевой оплаты труда (оклад, стимулирующие выплаты, районный коэффициент). Величина расходов по заработной плате определяется исходя из трудоемкости выполняемых работ и действующей системы окладов и тарифных ставок. В состав основной заработной платы включается премия, выплачиваемая ежемесячно из фонда заработной платы в размере 20 –30 % от тарифа или оклада. Для студента-дипломника основную заработную плату составляет государственная стипендия с учетом районного коэффициента.

Для расчета основной заработной платы (таблица 3.7) необходимо привести действительный годовой фонд рабочего времени руководителя и студента (таблица 3.6).

Таблица 3.6 – Баланс времени

Показатели рабочего времени	Руководитель	Студент-дипломник
Календарное число дней	365	365
Количество нерабочих дней (выходные дни и праздничные дни)	52 и 10	108 и 10
Потери рабочего времени (отпуск, больничные), дни	48	24
Действительный годовой фонд рабочего времени, дни	255	223

Статья включает основную заработную плату работников, непосредственно занятых выполнением НИИ, и дополнительную заработную плату:

$$Z_{ЗП} = Z_{осн} + Z_{доп}, \quad (3.6)$$

где  $Z_{осн}$  – основная заработная плата,  $Z_{доп}$  – дополнительная заработная плата (10% от основной).

Основная заработная плата руководителя (лаборанта, инженера) от предприятия рассчитывается по следующей формуле:

$$Z_{\text{осн}} = Z_{\text{дн}} + T_p, \quad (3.7)$$

где  $Z_{\text{осн}}$  – основная заработная плата одного работника,  $Z_{\text{дн}}$  – среднедневная заработная плата работника, руб.,  $T_p$  – продолжительность работ, выполняемых работником, раб.дн.

Среднедневная зарплата рассчитывается по формуле:

$$Z_{\text{дн}} = \frac{Z_m * M}{F_d}, \quad (3.8)$$

где  $Z_m$  – месячный должностной оклад работника, руб.,  $M$  – количество месяцев работы без отпуска в течение года: при отпуске в 48 раб. дней  $M = 10,4$  месяца, 6-дневная неделя,  $F_d$  – действительный годовой фонд рабочего времени научно-технического персонала, раб. дн. ( $F_d = 251$ ).

Месячный должностной оклад работника:

$$Z_m = Z_{\text{тс}}(1 + k_{\text{пр}} + k_d)k_p, \quad (3.9)$$

где  $Z_{\text{тс}}$  – заработная плата по тарифной ставке, руб.,  $k_{\text{пр}}$  – премиальный коэффициент ( $k_{\text{пр}} = 0.3$ ),  $k_d$  – коэффициент доплат и надбавок ( $k_{\text{пр}} = 0.2 - 0.5$ ),  $k_p$  – районный коэффициент ( $k_{\text{пр}} = 1.3$  для Томска).

Расчет основной платы представлен в таблице 3.7.

Таблица 3.7 – Расчет основной заработной платы

Исполнитель	Тарифная заработная плата, руб.	Районный коэффициент	Месячный должностной оклад работника, руб.	Среднедневная заработная плата, руб.	Продолжительность работ, дни	Заработная плата основная, руб.
Руководитель	23265	1,3	30244,5	1218	43	52374
<b>Итого:</b>						52374

### 3.4.3 Дополнительная заработная плата

Затраты по дополнительной заработной плате исполнителей темы учитывают величину предусмотренных Трудовым кодексом РФ доплат за

отклонение от нормальных условий труда, а также выплат, связанных с обеспечением гарантий и компенсаций (при исполнении государственных и общественных обязанностей, при совмещении работы с обучением, при предоставлении ежегодного оплачиваемого отпуска и т.д.).

Расчет дополнительной заработной платы ведется по следующей формуле:

$$Z_{\text{доп}} = k_{\text{доп}} * Z_{\text{осн}} = 0.15 * 52374 = 7856.1,$$

где  $Z_{\text{осн}}$  – основная заработная плата,  $k_{\text{доп}}$  – коэффициент дополнительной заработной платы на стадии проектирования ( $k_{\text{доп}} = 0.15$ ).

#### 3.4.4 Отчисления во внебюджетные фонды

В данной статье расходов отражаются обязательные отчисления по установленным законодательством Российской Федерации нормам органами государственного социального страхования (ФСС), пенсионного фонда (ПФ) и медицинского страхования (ФФОМС) от затрат на оплату труда работников.

Величина отчислений во внебюджетные фонды определяется исходя из следующей формулы:

$$Z_{\text{внеб}} = k_{\text{внеб}} * (Z_{\text{осн}} + Z_{\text{доп}}), \quad (3.10)$$

где  $k_{\text{внеб}}$  – коэффициент отчислений на уплату во внебюджетные фонды (пенсионный фонд, фонд обязательного медицинского страхования и пр.).

Для учреждений, осуществляющих образовательную и научную деятельность в 2019г., водится пониженная ставка 28% (п. 6 ч. 1 ст. 58 Закона 212-ФЗ).

Расчет отчислений во внебюджетные фонды приведен в таблице 3.8.

Таблица 3.8 – Отчисления во внебюджетные фонды

Исполнитель	Основная заработная плата, руб.	Дополнительная заработная плата, руб.
Руководитель проекта	52374	7856,1
Коэффициент отчисления во внебюджетные фонды	14664,72	2199,7

<b>Итого</b>	16864,42
--------------	----------

### 3.4.5 Прочие прямые затраты

К данному виду затрат относятся затраты на электроэнергию. Для юридических лиц стоимость 1 кВт·ч составляет 5,8 рублей. При умеренном пользовании ноутбук средней мощности потребляет 100 Вт в час в среднем. В день на работу затрачивается 6 часов, всего на работу с компьютером и оборудованием затрачивается 34 дня у инженера и 9 дней у руководителя. Тогда затраты на электроэнергию составят:

$$Z_{эн} = 100 * \frac{5,8}{1000} * 6 * 43 = 149.64 \text{ руб.}$$

### 3.4.6 Накладные расходы

Накладные расходы учитывают прочие затраты организации, не попавшие в предыдущие статьи расходов: печать и ксерокопирование материалов исследования, оплата услуг связи, электроэнергии, почтовые и телеграфные расходы, размножение материалов и т.д. Их величина определяется по следующей формуле:

$$C_{накл} = k_{накл}(Z_{осн} + Z_{доп}), \quad (3.11)$$

где  $k_{накл}$  – коэффициент накладных расходов, 16 %.

Получим:

$$Z_{накл} = 0,16 \cdot (48681.98 + 5841.84) = 8723.81 \text{ руб.}$$

### 3.4.7 Формирование бюджета затрат научно-исследовательского проекта

Рассчитанная величина затрат научно-исследовательской работы является основой для формирования бюджета затрат проекта, который при формировании договора с заказчиком защищается научной организацией в качестве нижнего предела затрат на разработку научно-технической продукции.

Определение бюджета затрат на научно-исследовательский проект приведено в таблице 3.9.

Таблица 3.9 – Расчет бюджета затрат

Наименование статьи	Сумма, руб.
1. Затраты на специальное оборудование	16 000
2. Затраты по основной заработной плате исполнителей проекта	52 374
3. Затраты по дополнительной заработной плате исполнителей проекта	7856
4. Отчисления во внебюджетные фонды	16864
5. Накладные расходы	14 895
<b>Бюджет затрат НИИ</b>	<b>107 989</b>

### 3.5 Определение ресурсной, финансовой, бюджетной, социальной и экономической эффективности исследования

Определение эффективности происходит на основе расчета интегрального показателя эффективности научного исследования. Его нахождение связано с определением двух средневзвешенных величин: финансовой эффективности и ресурсоэффективности.

Интегральный финансовый показатель разработки определяется как:

$$I_{\text{финр}}^{\text{исп.}i} = \frac{\Phi_p^i}{\Phi_{\text{max}}}, \quad (3.12)$$

где  $I_{\text{финр}}^{\text{исп.}i}$  – интегральный финансовый показатель разработки,  $\Phi_p^i$  – стоимость  $i$ -го варианта исполнения,  $\Phi_{\text{max}}$  – максимальная стоимость исполнения научно-исследовательского проекта.

$\Phi_{\text{max}}$  зависит от сложности проекта для которого разрабатывается ПО.

Т.к. стоимость всех вариантов исполнения одинакова, интегральные финансовые показатели также будут одинаковы и равны 1.

Интегральный показатель ресурсоэффективности вариантов исполнения объекта исследования можно определить следующим образом:

$$I_p^i = \sum a_i b_i, \quad (3.13)$$

где  $a_i$  – весовой коэффициент  $i$ -го варианта исполнения разработки,  $b_i$  – бальная оценка  $i$ -го варианта исполнения разработки, устанавливается экспертным путем по выбранной шкале оценивания.

Интегральный показатель эффективности вариантов исполнения разработки определяется на основании интегрального показателя ресурсоэффективности и интегрального финансового показателя по формулам:

$$I_{\text{исп1}} = \frac{I_{p-\text{исп1}}}{I_{\text{финр}}^{\text{исп1}}}, \quad (3.14)$$

$$I_{\text{исп2}} = \frac{I_{p-\text{исп2}}}{I_{\text{финр}}^{\text{исп2}}},$$

Так как интегральные финансовые показатели одинаковы и равны 1, то интегральные показатели эффективности вариантов исполнения разработки равны соответствующим интегральным показателям ресурсоэффективности.

Сравнение интегрального показателя эффективности вариантов исполнения разработки позволит определить сравнительную эффективность проекта и выбрать наиболее целесообразный вариант из предложенных.

Сравнительная эффективность проекта:

$$\mathcal{E}_{\text{ср}} = \frac{I_{\text{исп}i}}{I_{\text{исп1}}}. \quad (3.15)$$

Было рассмотрено два варианта исполнения алгоритма. На основании этого необходимо провести сравнительную характеристику вариантов исполнения (таблица 3.10).

Таблица 3.10– Сравнительная оценка характеристик вариантов исполнения проекта

Критерии	Весовой коэффициент	Исп. 1	Исп. 2
1. Повышение производительности труда	0,3	5	5
2. Удобство в эксплуатации	0,2	4	3
3. Удобство в считывании исходных данных	0,2	5	2
4. Скорость работы	0,1	4	3
5. Простота эксплуатации	0,1	4	3



6. Техническая поддержка платформы	0,1	5	3
$I_p^i$		<b>4,6</b>	<b>3,4</b>

На основании полученных показателей выполним сравнение интегрального показателя эффективности вариантов исполнения разработки (таблица 3.11).

Таблица 3.11 – Сравнительная эффективность разработки

Показатели	Исп.1	Исп.2
Интегральный финансовый показатель разработки	1	1
Интегральный показатель ресурсоэффективности разработки	4,6	3,4
Интегральный показатель эффективности	4,6	3,4
Сравнительная эффективность вариантов исполнения	1	0,74

С позиции финансовой и ресурсной эффективности на основании таблицы, первый вариант исполнения системы наиболее выгодный. Данный вариант исполнения и используется в выпускной квалификационной работе.

**ЗАДАНИЕ ДЛЯ РАЗДЕЛА  
«СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ»**

Студенту:

<b>Группа</b>	<b>ФИО</b>
8Б51	Кулигин Сергей Михайлович

<b>Школа</b>	<b>ИШИТР</b>	<b>Отделение (НОЦ)</b>	<b>Информационных технологий</b>
<b>Уровень образования</b>	Бакалавриат	<b>Направление/специальность</b>	Прикладная математика и информатика

Тема ВКР:

«Решение задачи квадратичного программирования на основе преобразования Гаусса-Жордана»

**Исходные данные к разделу «Социальная ответственность»:**

1. Характеристика объекта исследования (вещество, материал, прибор, алгоритм, методика, рабочая зона) и области его применения.	1. Алгоритм решения задач квадратичного программирования на основе преобразования Гаусса-Жордана. Область применения: автоматические системы управления, оценка и обработка сигналов.
---	---

**Перечень вопросов, подлежащих исследованию, проектированию и разработке:**

<p><b>1. Производственная безопасность</b></p> <p>1.1. Анализ выявленных вредных факторов при разработке и эксплуатации проектируемого решения в следующей последовательности.</p> <p>1.2. Анализ выявленных опасных факторов при разработке и эксплуатации проектируемого решения в следующей последовательности.</p>	<p>1.1 Рассмотрены вредные факторы: Отклонение показателей микроклимата рабочей зоны; повышенный уровень шума на рабочем месте; отсутствие или недостаток естественного света, а также недостаточная освещенность рабочей зоны.</p> <p>1.2 Рассмотрены опасные факторы: повышенное значение напряжения в электрической цепи, замыкание которой может произойти через тело человека; возгорание эксплуатируемого оборудования.</p>
<p><b>2. Экологическая безопасность:</b></p> <p>2.1 Анализ воздействия объекта на окружающую среду;</p> <p>2.2 Разработать решения по обеспечению экологической безопасности со ссылками на НТД по охране окружающей среды.</p>	<p>2.1 Рассмотрены негативно влияющие на экологию факторы при эксплуатации компьютера.</p> <p>2.2 Решения по обеспечению экологической безопасности согласно нормативным документам.</p>
<p><b>3. Безопасность в чрезвычайных ситуациях:</b></p> <p>3.1 Перечень возможных ЧС при разработке и эксплуатации проектируемого решения;</p> <p>3.2 Разработка действий в результате возникшей ЧС и мер по ликвидации её последствий.</p>	<p>3.1 Перечень возможных ЧС, которые могут возникнуть при работе в помещении офиса.</p> <p>3.2 Способы защиты от пожара и ликвидация последствий.</p>
<p><b>4. Правовые и организационные вопросы обеспечения безопасности:</b></p> <p>4.1 Организационные мероприятия при компоновке рабочей зоны.</p>	<p>4.1 Организационные мероприятия по обеспечению безопасности трудящихся за персональным компьютером.</p>

**Дата выдачи задания для раздела по линейному графику**

**Задание выдал консультант:**

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Ассистент ООТД	Мезенцева Ирина Леонидовна			

**Задание принял к исполнению студент:**

Группа	ФИО	Подпись	Дата
8Б51	Кулигин Сергей Михайлович		

## Введение

Выпускная квалификационная работа представляет собой сравнительное исследование моделей параметризации результатов тестирования. Во время выполнения данной работы были разработаны и запрограммированы алгоритмы реализации параметрических моделей для оценки учебных достижений студентов посредством тестов. Все расчеты и алгоритмы реализованы в среде Visual Studio на языке программирования C++.

Данная работа имеет высокий потенциал. Описанный в работе алгоритм может использоваться для, например, решения задач о распределении ограниченных ресурсов. Данный способ имеет ряд преимуществ перед другими применяемыми методами таких, как метод Баранкина-Дорфмана и метод с использованием операторов проекторов.

Предполагаемое место работы – компьютерный класс Кибернетического центра ТПУ. Основные средства работы – персональный компьютер и локальная вычислительная сеть с выходом в Интернет. Все действия, описанные в данной выпускной квалификационной работе, были совершены в городе Томск, на территории кампуса томского политехнического университета.

## **4 Социальная ответственность**

### **4.1 Правовые и организационные вопросы обеспечения безопасности**

#### **4.1.1 Специальные правовые нормы трудового законодательства**

Как уже было неоднократно отмечено, при работе с персональным компьютером очень важную роль играет соблюдение правильного режима труда и отдыха. В противном случае у персонала отмечаются значительное напряжение зрительного аппарата с появлением жалоб на неудовлетворенность работой, головные боли, раздражительность, нарушение сна, усталость и болезненные ощущения в глазах, в пояснице, в области шеи и руках.

При восьми часовой рабочей смене на ВДТ и ПЭВМ перерывы в работе должны составлять от 10 до 20 минут каждые два часа работы. В перерывах, рекомендуется проводить комплекс упражнений для глаз.

По типовой инструкции по охране труда при работе на персональном компьютере ТОИ Р-45-084-01 всё же вменяется установление перерывов. В целом продолжительность непрерывной работы за компьютером не должна превышать 2-х часов.

Следует обратить внимание на то, что основная работа за компьютером предусматривает не менее 50 % времени в течение рабочей смены или рабочего дня нахождения за ним. Время перерыва зависит от вида и сложности осуществляемой работы путем деления на группы. Выделяют 3 группы: А (работа по считыванию информации с экрана компьютера с предварительным запросом), Б (работа по вводу информации), В (творческая работа в режиме диалога с компьютером).

В целях обеспечения прав и свобод человека и гражданина работодатель и его представители при обработке персональных данных

работника обязаны соблюдать следующие общие требования:

1. обработка персональных данных работника может осуществляться исключительно в целях обеспечения соблюдения законов и иных нормативных правовых актов, содействия работникам в трудоустройстве, получении образования и продвижении по службе, обеспечения личной безопасности работников, контроля количества и качества выполняемой работы и обеспечения сохранности имущества;

2. при определении объема и содержания обрабатываемых персональных данных работника работодатель должен руководствоваться Конституцией Российской Федерации, настоящим Кодексом и иными федеральными законами;

3. все персональные данные работника следует получать у него самого;

4. работодатель не имеет права получать и обрабатывать сведения о работнике, относящиеся в соответствии с законодательством Российской Федерации в области персональных данных к специальным категориям персональных данных, за исключением случаев, предусмотренных настоящим Кодексом и другими федеральными законами;

5. работодатель не имеет права получать и обрабатывать персональные данные работника о его членстве в общественных объединениях или его профсоюзной деятельности, за исключением случаев, предусмотренных настоящим Кодексом или иными федеральными законами;

6. при принятии решений, затрагивающих интересы работника, работодатель не имеет права основываться на персональных данных работника, полученных исключительно в результате их автоматизированной обработки или электронного получения;

7. защита персональных данных работника от неправомерного их использования или утраты должна быть обеспечена работодателем за счет

его средств в порядке, установленном настоящим Кодексом и иными федеральными законами;

8. работники и их представители должны быть ознакомлены под роспись с документами работодателя, устанавливающими порядок обработки персональных данных работников, а также об их правах и обязанностях в этой области;

9. работники не должны отказываться от своих прав на сохранение и защиту тайны;

10. работодатели, работники и их представители должны совместно вырабатывать меры защиты персональных данных работников.

Согласно ТК РФ Статья 135 об установлении заработной платы, заработная плата работнику устанавливается трудовым договором в соответствии с действующими у данного работодателя системами оплаты труда.

Что же касается нормы выработки, времени, нормативы численности и других норм – они устанавливаются в соответствии с достигнутым уровнем техники, технологии, организации производства и труда. Достижение высокого уровня выработки продукции (оказания услуг) отдельными работниками за счет применения по их инициативе новых приемов труда и совершенствования рабочих мест не является основанием для пересмотра ранее установленных норм труда.

Конкретные размеры повышения оплаты труда устанавливаются работодателем с учетом мнения представительного органа работников в порядке, установленном статьей 372 настоящего Кодекса для принятия локальных нормативных актов, либо коллективным договором, трудовым договором.

#### **4.1.2 Организационные мероприятия при компоновке рабочей зоны**

Под проектированием рабочего места понимается целесообразное пространственное размещение в горизонтальной и вертикальной плоскостях

функционально взаимосвязанных средств производства, необходимых для осуществления трудового процесса. При проектировании рабочих мест должны быть учтены освещенность, температура, влажность, давление, шум, наличие вредных веществ, электромагнитных полей и другие санитарно-гигиенические требования к организации рабочих мест.

Площадь одного рабочего места с компьютером должна быть не менее 6 м<sup>2</sup>. При размещении рабочих мест с персональными компьютерами должны учитываться расстояния между рабочими столами с мониторами.

Для комфортной работы стол инженера-программиста должен удовлетворять следующим условиям:

- высота стола должна быть выбрана с учетом возможности сидеть свободно, в удобной позе, при необходимости опираясь на подлокотники;
- нижняя часть стола должна быть сконструирована так, чтобы программист мог удобно сидеть, не был вынужден поджимать ноги;
- поверхность стола должна обладать свойствами, исключающими появление бликов в поле зрения программиста;
- конструкция стола должна предусматривать наличие выдвижных ящиков;
- высота рабочей поверхности рекомендуется в пределах 680-760мм;
- Высота поверхности, на которую устанавливается клавиатура, должна быть около 650мм.

Помещения с компьютерами в обязательном порядке должны быть оборудованы системами эффективной приточно-вытяжной вентиляцией отопления и кондиционирования воздуха. Внутренняя отделка интерьера помещений с компьютерами должна быть сделана при использовании диффузно-отражающих материалов. В обязательном порядке в помещении должны находиться углекислотный огнетушитель для тушения пожара и аптечка первой медицинской помощи.

Рабочее место в комнате № 104 КЦ ТПУ отвечает данным условиям.



## 4.2 Профессиональная социальная безопасность

В данном подразделе производится анализ вредных и опасных факторов, которые могут возникнуть на рабочем месте при проведении исследований и использовании результатов исследования.

Опасным производственным фактором (ОПФ) называется такой производственный фактор, воздействие которого на работающего мгновенно приводит к травме или летальному исходу. Травма – это повреждение тканей организма и нарушение его функций внешним воздействием.

Вредным производственным фактором (ВПФ) называется такой производственный фактор, воздействие которого на работающего в определенных условиях приводит к заболеванию или снижению трудоспособности.

Выявленные факторы представлены в таблице 4.1.

Таблица 4.1 – Опасные и вредные факторы при проведении сравнительного исследования результатов оценивания учебных достижений

Источник фактора, наименование видов работ	Факторы		Нормативные документы
	Вредные	Опасные	
– Работа за персональной электронно-вычислительной машиной в компьютерной аудиотории.	– Отклонение показателей микроклимата рабочей зоны; – отсутствие или недостаток естественного света, а также недостаточная освещенность рабочей зоны; – повышенный уровень шума на рабочем месте;	– Повышенное значение напряжения в электрической цепи, замыкание которой может произойти через тело человека; возгорание эксплуатируемого оборудования.	– Гигиенические требования к микроклимату производственных помещений, СанПиН 2.2.4-548-96; – Гигиенические требования к естественному, искусственному и совмещённому освещению жилых и общественных зданий, СанПиН 2.2.1/2.1.1.1278–03 – Шум. Общие требования безопасности, СН 2.2.4/2.1.8.562–96. – Гигиенические требования к персональным электронно-вычислительным машинам и организации

			работы, СанПиН 2.2.2/2.4.1340-03; – Электробезопасность, ГОСТ 12.1.038–82 ССБТ
--	--	--	---

#### **4.2.1 Анализ вредных и опасных факторов**

##### ***Производственный шум***

Шум – колебания различной физической природы, отличающиеся сложностью спектральной и временной структуры. Шум создает значительную нагрузку на нервную систему человека, оказывая на него психологическое воздействие. Шум способен замедлять реакцию человека и угнетать центральную нервную систему, вызывая изменения скорости пульса и дыхания, а также провоцирует возникновение сердечно - сосудистых заболеваний [13].

Человек, постоянно подвергающийся воздействию шума, быстро переутомляется, чаще страдает от слабости и головокружения. Уровень звука на рабочих местах, связанных с творческой деятельностью, научной деятельностью, программированием, преподаванием и обучением не должен превышать 50 дБА согласно СН 2.2.4/2.1.8.562–96 [13].

Меры, которые необходимо принять, для того чтобы помещение было менее зашумленным – это обеспечить нормальную вентиляцию системного блока. Для охлаждения необходимо оборудовать со стороны вентиляционных отверстий хотя бы 20-30 см свободного пространства.

##### ***Микроклимат рабочего помещения***

Как центральное отопление, так и обогреватели сушат воздух. Пересушенный воздух при критических показателях создает неоспоримую опасность для здоровья человека: способствует возникновению инфекций, обострение аллергических заболеваний и астмы.

В помещениях, где установлены компьютеры, должны соблюдаться определенные параметры микроклимата. Эти нормы устанавливаются в зависимости от времени года, характера трудового процесса и характера

производственного помещения. Оптимальные и допустимые параметры микроклимата и нормы подачи свежего воздуха в помещения приведены на рисунках 4.1 и 4.2 [14].

Период года	Категория работ по уровням энергозатрат, Вт	Температура воздуха, °С	Температура поверхностей, °С	Относительная влажность воздуха, %	Скорость движения воздуха, м/с
Холодный	Ia (до 139)	22 - 24	21 - 25	60 - 40	0,1
	Iб (140 - 174)	21 - 23	20 - 24	60 - 40	0,1
	IIa (175 - 232)	19 - 21	18 - 22	60 - 40	0,2
	IIб (233 - 290)	17 - 19	16 - 20	60 - 40	0,2
	III (более 290)	16 - 18	15 - 19	60 - 40	0,3
Теплый	Ia (до 139)	23 - 25	22 - 26	60 - 40	0,1
	Iб (140 - 174)	22 - 24	21 - 25	60 - 40	0,1
	IIa (175 - 232)	20 - 22	19 - 23	60 - 40	0,2
	IIб (233 - 290)	19 - 21	18 - 22	60 - 40	0,2
	III (более 290)	18 - 20	17 - 21	60 - 40	0,3

Рисунок 4.1 – Оптимальные величины показателей микроклимата на рабочих местах производственных помещений

Период года	Категория работ по уровню энергозатрат, Вт	Температура воздуха, °С		Температура поверхностей, °С	Относительная влажность воздуха, %	Скорость движения воздуха, м/с	
		диапазон ниже оптимальных величин	диапазон выше оптимальных величин			для диапазона температур воздуха ниже оптимальных величин, не более	для диапазона температур воздуха выше оптимальных величин, не более <*>
Холодный	Ia (до 139)	20,0 - 21,9	24,1 - 25,0	19,0 - 26,0	15 - 75 <*>	0,1	0,1
	Iб (140 - 174)	19,0 - 20,9	23,1 - 24,0	18,0 - 25,0	15 - 75	0,1	0,2
	IIa (175 - 232)	17,0 - 18,9	21,1 - 23,0	16,0 - 24,0	15 - 75	0,1	0,3
	IIб (233 - 290)	15,0 - 16,9	19,1 - 22,0	14,0 - 23,0	15 - 75	0,2	0,4
	III (более 290)	13,0 - 15,9	18,1 - 21,0	12,0 - 22,0	15 - 75	0,2	0,4
Теплый	Ia (до 139)	21,0 - 22,9	25,1 - 28,0	20,0 - 29,0	15 - 75 <*>	0,1	0,2
	Iб (140 - 174)	20,0 - 21,9	24,1 - 28,0	19,0 - 29,0	15 - 75 <*>	0,1	0,3
	IIa (175 - 232)	18,0 - 19,9	22,1 - 27,0	17,0 - 28,0	15 - 75 <*>	0,1	0,4
	IIб (233 - 290)	16,0 - 18,9	21,1 - 27,0	15,0 - 28,0	15 - 75 <*>	0,2	0,5
	III (более 290)	15,0 - 17,9	20,1 - 26,0	14,0 - 27,0	15 - 75 <*>	0,2	0,5

<\*> При температурах воздуха 25 °С и выше максимальные величины относительной влажности воздуха должны приниматься в соответствии с требованиями п. 6.5.

<\*> При температурах воздуха 26 - 28 °С скорость движения воздуха в теплый период года должна приниматься в соответствии с требованиями п. 6.6.

Рисунок 4.2 – Допустимые величины показателей микроклимата на рабочих местах производственных помещений

### ***Освещенность рабочего помещения***

Соответствующее производственное освещение способствует улучшению условий зрительной работы, как следствие снижает утомляемость и способствует повышению производительности труда.

Существует три вида освещения – естественное солнечное освещение, искусственное и комбинированное.

В компьютерных залах должно быть естественное и искусственное освещение, то есть комбинированное. Естественное освещение создается прямыми солнечными лучами. Искусственное освещение в помещениях эксплуатации компьютеров должно осуществляться системой общего равномерного освещения. Местное освещение не должно создавать бликов на поверхности экрана и увеличивать освещенность экрана более 300 лк согласно СНиП 23-05-95 [17].

Для оценки использования естественного света введено понятие коэффициента естественной освещенности (КЕО) и установлены минимальные допустимые значения КЕО — это отношение освещенности  $E_B$  внутри помещения за счет естественного света к наружной освещенности  $E_H$  от всей полусферы небосклона, выраженное в процентах:

$$КЕО = \left(\frac{E_B}{E_H}\right) 100\%.$$

По характеристике зрительской работы труд учащихся можно отнести ко второму разряду работы, и при боковом естественном освещении в аудитории, лаборатории на рабочих столах и партах должен обеспечиваться  $КЕО = 1,5\%$ . К средствам нормализации освещения производственных помещений и рабочих мест относятся: источники света, осветительные приборы, световые проемы, светозащитные устройства, светофильтры.

### ***Электрический ток***

В связи с тем, что для работы компьютера используется электрическая энергия, его эксплуатация должна соответствовать правилам техники безопасности при эксплуатации электроустановок потребителей. С этой

точки зрения компьютер является потенциальным источником опасности поражения человека электрическим током.

Проходя через организм человека, электрический ток оказывает термическое, электролитическое и биологическое действие. Первое заключается в нагреве и ожогах различных участков тела человека, второе — в изменении состава и свойств крови.

Чтобы избежать поражения электрическим током, необходимо выполнять правила, указанные в [18]:

#### **4.2.2 Обоснование мероприятий по защите исследователя от действия опасных и вредных факторов**

Большое значение для профилактики статических физических перегрузок имеет правильная организация рабочего места человека, работающего с компьютером. Рабочее место должно быть организовано в соответствии с требованиями стандартов, технических условий и методических указаний по безопасности труда.

Невыполнение требований к расположению и компоновке рабочего места может привести к получению работником производственной травмы или развития у него профессионального заболевания. Рабочее место программиста должно соответствовать требованиям СанПин 2.2.2/2.4.1340-03. Конструкция оборудования и рабочего места при выполнении работ в положении сидя должна обеспечивать оптимальное положение работающего, которое достигается регулированием высоты рабочей поверхности, высоты сидения, оборудованием пространства для размещения ног и высотой подставки для ног. Схемы размещения рабочих мест с персональными компьютерами должны учитывать расстояния между рабочими столами с мониторами: расстояние между боковыми поверхностями мониторов не менее 1,2 м, а расстояние между экраном монитора и тыльной частью другого монитора не менее 2,0 м. Клавиатура должна располагаться на поверхности стола на расстоянии 100-300 мм от

края, обращенного к пользователю. Быстрое и точное считывание информации обеспечивается при расположении плоскости экрана ниже уровня глаз пользователя, предпочтительно перпендикулярно к нормальной линии взгляда (нормальная линия взгляда 15 градусов вниз от горизонтали). Рабочие места с ПЭВМ при выполнении творческой работы, требующей значительного умственного напряжения или высокой концентрации внимания, рекомендуется изолировать друг от друга перегородками высотой 1,5 - 2,0 м.

Разрабатываемый в ходе выполнения ВКР программный продукт используется для анализа численных данных и наборов значений. При работе с данным продуктом не требуется постоянное его использование, так как программа анализирует входные данные и получает необходимый результат, который, непосредственно, будет использован в дальнейшем. Работнику необходимо контролировать параметры входных данных и фиксировать выходные параметры. Следовательно, преимуществом данного продукта является практически полная автономность.

### **4.3 Экологическая безопасность**

Рассмотрим загрязнения литосферы в результате исследовательской деятельности бытовым мусором, на примере люминесцентных ламп. Их эксплуатация требует осторожности и четкого выполнения инструкции по обращению с данным отходом (код отхода 35330100 13 01 1, класс опасности – 1). В данной лампе содержится опасное вещество ртуть в газообразном состоянии. При не правильной утилизации, лампа может разбиться и пары ртути могут попасть в окружающую среду. Вдыхание паров ртути может привести к тяжелому повреждению здоровья.

При перегорании ртутьсодержащей лампы (выходе из строя) её замену осуществляет лицо, ответственное за сбор и хранение ламп (обученное по электробезопасности и правилам обращения с отходом).

Отработанные люминесцентные лампы сдаются только на полигон токсичных отходов для захоронения. Запрещается сваливать отработанные люминесцентные лампы с мусором.

Бытовой мусор помещений организаций несортированный, образованный в результате деятельности работников предприятия (код отхода 91200400 01 00 4). Агрегатное состояние отхода твердое; основные компоненты: бумага и древесина, металлы, пластмассы и др. Для сбора мусора рабочее место оснащается урной. При заполнении урны, мусор выносится в контейнер бытовых отходов. Предприятие заключает договор с коммунальным хозяйством по вывозу и размещению мусора на организованных свалках.

#### **4.4 Безопасность в чрезвычайных ситуациях**

##### **4.4.1 Анализ вероятных ЧС**

Наиболее вероятной ЧС в рамках рассматриваемого помещения является пожар. Помещение, в котором велась работа по степени пожаробезопасности относится к категории Г (умеренная пожароопасность), т.е. к помещению, в котором находятся негорючие вещества и материалы в горячем, раскаленном или расплавленном состоянии.

Гипотетически возникновение пожара может возникнуть от следующих источников воспламенения [19]:

- искра при разряде статистического электричества;
- искра от электрооборудования;
- открытое пламя.

Также на рабочем месте запрещается иметь огнеопасные вещества и выполнять следующие действия [19]:

- курить;
- зажигать огонь;

- включать электрооборудование, если в помещении пахнет газом;
- сушить что-либо на отопительных приборах;
- закрывать вентиляционные отверстия в электроаппаратуре.

#### **4.4.2 Обоснование мероприятий по предотвращению ЧС и разработка порядка действия в случае возникновения ЧС**

К мерам по предупреждению пожара отнесем следующие пожарно-профилактические мероприятия:

- соблюдение эксплуатационных норм оборудования;
- обучение персонала правилам техники безопасности;
- издание противопожарных инструкций, планов эвакуации.

Основными мерами по повышению устойчивости помещения к данной ЧС являются в первую очередь исключение образования благоприятной для пожара среды, а также использование трудно сгораемых материалов при отделке рабочего помещения [19].

Необходимо предусмотреть безопасную эвакуацию людей на случай возникновения пожара. При пожаре люди должны покинуть помещение в течение минимального времени. Помещение, в котором выполнялась работа, входит в общий план эвакуации этажа, который предусматривает выход из всех помещений этажа в основной или запасной эвакуационные выходы здания. Эвакуация проводится согласно плану эвакуации, который выставлен на всеобщее обозрение в нескольких местах на каждом этаже (рисунок 4.1).

В каждом кабинете установлен углекислотный огнетушитель ОУ-2 и табличка с указанием лица, ответственного за пожарную безопасность.



ПЛАН  
эвакуации сотрудников и студентов  
кафедры ПМ ИК в случае пожара в корпусе ИК - 1 этаж

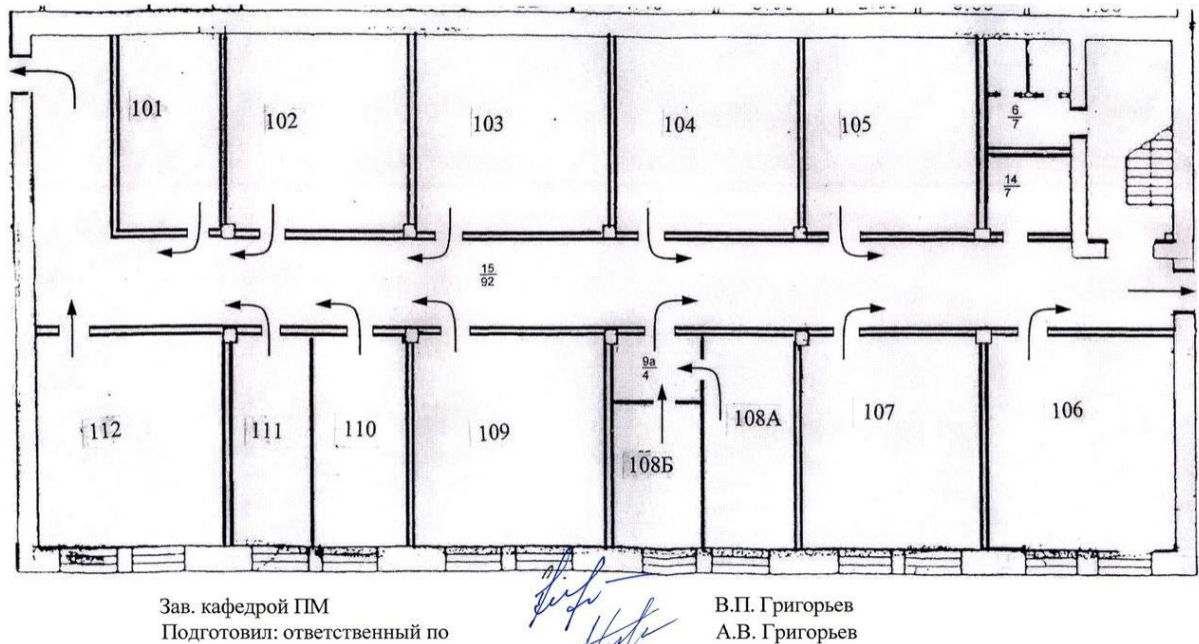


Рисунок 4.1 – План эвакуации

Необходимые действия в результате возникшей ЧС и меры по ликвидации её последствий указаны в [19].

### Выводы и рекомендации

Проанализировав условия труда на рабочем месте, где была разработана бакалаврская работа, можно сделать вывод, что помещение удовлетворяет необходимым нормам и в случае соблюдения техники безопасности и правил пользования компьютером работа в данном помещении не приведет к ухудшению здоровья работника. Само помещение и рабочее место в нем удовлетворяет всем нормативным требованиям. Кроме того, действие вредных и опасных факторов сведено к минимуму, т.е. микроклимат, освещение и электробезопасность соответствуют требованиям.

Относительно рассмотренного вопроса об экологической безопасности можно сказать, что деятельность в ходе выполнения выпускной квалификационной работы не представляет опасности

окружающей среды.

## Заключение

В ходе выполнения преддипломной практики были получены следующие результаты:

- проведён обзор литературы по тематике математического программирования, в частности, квадратичного программирования;
- сформулирована актуальность поставленной задачи;
- сформулированы концептуальная и математическая постановки задачи;
- сформулированы выражения, позволяющие перейти от задачи квадратичного программирования со смешанными ограничениями к задаче только с ограничениями-неравенствами;
- разработана модель алгоритма решения задачи квадратичного программирования для заданного целевого функционала с наложенными ограничениями;
- составлены алгоритм преобразования целевого функционала с учётом ограничений-равенств и алгоритм поиска искомого решения поставленной задачи;
- разработана программа на объектно-ориентированном языке C++ в среде Visual Studio, реализующая составленный алгоритм учёта ограничений-равенств и метод Хилдрета-Д'Эзопо.
- проверена и подтверждена работоспособность программы на тестовом примере.

### Список использованных источников

1. Кюнци Г. П. Нелинейное программирование / Г. П. Кюнци, В. Крелле. – М.: Изд-во «Советское радио», 1965. – 303 с.
2. Данциг Дж. Линейное программирование его применения и обобщения / Дж. Данциг. – М.: Изд-во «Прогресс», 1966. – 600 с.
3. Вылегжанин О.Н., Шкатова Г.И. Учет ограничений равенств при решении оптимизационных задач с линейными ограничениями // Известия Томского политехнического университета. – 2008. – Т. 312. – № 5. – С. 76–78.
4. Вылегжанин О.Н., Шкатова Г.И. Решение задачи линейного программирования с использованием оператора-проектора // Известия Томского политехнического университета. – 2009. – Т. 314. – № 5. – С. 37–40.
5. Гилл Ф., Мюррей Г., Райт М. Практическая оптимизация. – М.: Мир, 1985. – 509 с.
6. Цыганков А.А. Новые условия экстремума для гладких задач с ограничениями в форме равенств // Журнал вычислительной математики и математической физики. – 2001. – Т.41. – № 10. – С. 1474–1481.
7. Вылегжанин О.Н., Шкатова Г.И. Сравнительная оценка двух методов выбора наилучших линейных регрессоров // Применение математических методов и ЭВМ в методико-биологических исследованиях: Межвузовский научно-технический сборник. – Томск: Изд-во ТПУ, 1988. – С. 18–22.
8. Васильев Ф.П., Иваницкий А.Ю. Линейное программирование. – М.: Факториал, 1998. – 323 с.
9. Гантмахер Ф.Р. Теория матриц. – 5-е изд. – М.: Физматлит, 2004. – 559 с.
10. Вылегжанин О.Н. Сравнительный анализ возможностей линейного и квадратичного программирования в количественной спектрофотометрии смесей неполностью известного качественного состава // Журнал прикладной спектроскопии. – 1990. – Т. 52. – № 6. – С. 984–988.

11. Небаба С.Г., Вылегжанин О.Н. Построение и исследование алгоритма поиска первой крайней подсистемы для заданной совместной системы линейных неравенств // Известия Томского политехнического университета. – 2012. – Т. 320. – № 5. – С. 17–21.
12. Черников С. Н. Линейные неравенства. – М.: Наука, 1968. – 488 с.
13. СН 2.2.4/2.1.8.562–96. Шум на рабочих местах, в помещениях жилых, общественных зданий и на территории застройки.
14. СанПиН 2.2.548-96. Гигиенические требования к микроклимату производственных помещений.
15. СанПиН 2.2.2/2.4.1340-03 Гигиенические требования к персональным электронно-вычислительным машинам и организации работы.
16. Гигиенические требования к видеодисплейным терминалам, персональным электронно-вычислительным машинам и организации работы. Выписки из СанПиН 2.2.2.542-96.
17. СНиП\*23-05-95 Естественное и искусственное освещение.
18. ГОСТ 12.1.019 (с изм. №1) ССБТ. Электробезопасность. Общие требования и номенклатура видов защиты.
19. Федеральный закон Российской Федерации от 22 июля 2008 г. N 123-ФЗ "Технический регламент о требованиях пожарной безопасности".
20. Федеральный классификационный каталог отходов [Электронный ресурс] – 2013. – Режим доступа: <http://www.ecoguild.ru/>, свободный. – Загл. с экрана.
21. Об утверждении правил обращения с отходами производства и потребления в части осветительных устройств, электрических ламп, ненадлежащие сбор, накопление, использование, обезвреживание, транспортирование и размещение которых может повлечь причинение вреда жизни, здоровью граждан, вреда животным, растениям и окружающей среде: Постановление Правительства Российской Федерации от 3 сентября 2010 года № 681.

22.ТОИ Р-45-084-01. Типовая инструкция по охране труда при работе на персональном компьютере.

## Приложение А Листинг процедуры расчёта обратной матрицы

```
void inversion(double **A, int N)
{
    double temp;
    double **E = new double *[N];
    for (int i = 0; i < N; i++)
        E[i] = new double [N];
    for (int i = 0; i < N; i++){
        for (int j = 0; j < N; j++){
            E[i][j] = 0.0;
            if (i == j)
                E[i][j] = 1.0;}
    for (int k = 0; k < N; k++){
        temp = A[k][k];
        for (int j = 0; j < N; j++){
            A[k][j] /= temp;
            E[k][j] /= temp;}
        for (int i = k + 1; i < N; i++){
            temp = A[i][k];
            for (int j = 0; j < N; j++){
                A[i][j] -= A[k][j] * temp;
                E[i][j] -= E[k][j] * temp;}}}
    for (int k = N - 1; k > 0; k--){
        for (int i = k - 1; i >= 0; i--){
            temp = A[i][k];
            for (int j = 0; j < N; j++){
                A[i][j] -= A[k][j] * temp;
                E[i][j] -= E[k][j] * temp;}}}
    for (int i = 0; i < N; i++)
        for (int j = 0; j < N; j++)
            A[i][j] = E[i][j];
    for (int i = 0; i < N; i++)
        delete [] E[i];
    delete [] E;
}
```

## Приложение Б Листинг процедуры расчёта псевдообратной матрицы

```
void find_pseudo_inverse (double **C, double **CPlus, int z, int ma)
{
    double **Cstr = new double*[ma];
    for(int i=0; i<ma; ++i){
        Cstr[i] = new double[1];}
    double **CstrT = new double*[1];
    for(int i=0; i<1; ++i){
        CstrT[i] = new double[ma];}
    double **CstrTd = new double*[1];
    for(int i=0; i<1; ++i){
        CstrTd[i] = new double[1];}
    for(int i=0; i<ma; ++i){
        Cstr[i][0] = C[i][z];}
        for(int i=0; i<ma; ++i){
            CstrT[0][i] = C[i][z];}
    multiplication_matrix(CstrT, 1, ma, Cstr, ma, 1, CstrTd);
        for(int j=0; j<ma; ++j){
            CPlus[0][j]=CstrT[0][j]*(1/CstrTd[0][0]);}
}
```



## Приложение В Листинг процедуры умножения матриц

```
void multiplication_matrix (double **A, int ma, int na, double **B, int mb, int nb, double
**C)
{
    for (int i = 0; i < ma; i++) {
        for (int j = 0; j < nb; j++) {
            C[i][j] = 0;
            for (int j1 = 0; j1 < mb; j1++) {
                C[i][j] = C[i][j] + (A[i][j1] * B[j1][j]);}}}
}
```

## Приложение Г Листинг процедуры поиска столбца матрицы с максимальной нормой

```
void find_max_norm (double **A, int ma, int na, int &k, double &norm)
{
    double **a = new double*[ma];
    for(int i=0; i<ma; ++i){
        a[i] = new double[1];}
    double **aT = new double*[1];
    for(int i=0; i<1; ++i){
        aT[i] = new double[ma];}
    double **aP = new double*[1];
    for(int i=0; i<1; ++i){
        aP[i] = new double[1];}
    double summ = 0;
    norm = 0;
    for(int j=0; j<na; ++j){
        summ = 0;
        for(int i=0; i<ma; ++i){
            a[i][0] = A[i][j];
            aT[0][i] = a[i][0];}
        multiplication_matrix(aT, 1, ma, a, ma, 1, aP);
        summ = aP[0][0];
        if (norm < (sqrt(summ))) {
            norm = sqrt(summ);
            k = j;}}
}
```

## Приложение Д Листинг процедуры вывода матрицы

```
void print_matrix (double **A, int ma, int na, string text)
{
    cout<<endl<<endl<<text;
    for(int i=0; i<ma; ++i){
        for(int j=0; j<na; ++j){
            if (j==0) cout<<endl;
            cout<<A[i][j]<<" ";}}
}
```

## Приложение Е Листинг процедуры удаления столбца из одной матрицы и запись в другую

```
void column_swaps (double **A, double **A2, int ma, int na, double **An, double **A2n,
int ma2, int z, int k, double **P, double **Pn)
{
    //Перенос столбца с максимальной нормой из A1 в An
    for(int i=0; i<ma; ++i){
        An[i][z-1]=A[i][k];}
    //Перенос столбца с максимальной нормой из A2 в An
    for(int i=0; i<ma2; ++i){
        A2n[i][z-1]=A2[i][k];}
    //Удаление столбца с максимальной нормой из A1
    for(int j=k; j<na; ++j){
        for(int i=0; i<ma; ++i){
            if (j == na-1) {
                A[i][j] = 0;}
            else
                A[i][j] = A[i][j+1];}
    //Удаление столбца с номером столбца с максимальной нормой из A2
    for(int i=0; i<ma2; ++i){
        if (j == na-1) {
            A2[i][j] = 0;}
        else
            A2[i][j] = A2[i][j+1];}}
    //Перенос элемента с номером столбца с максимальной нормой из P в Pn
    Pn[z-1][0]=P[k][0];
    //Удаление элемента с номером столбца с максимальной нормой из P
    for(int i=k; i<na; ++i){
        if (i == na-1) {
            P[i][0] = 0;}
        else
            P[i][0] = P[i+1][0];}
}
```

## Приложение Ж Листинг тела программы

```
int _tmain(int argc, _TCHAR* argv[])
{
    int ma; int ma2; int na; int z = 0; int k = 0; double norm; double tol; bool check
= true;
    int t = 100; int qua = 0; int sum1; int sum2; int r; int l; int f1; int f2;
    freopen("input.txt","r",stdin);
    freopen("output.txt","w",stdout);
    scanf("%lf",&tol);
    scanf("%d",&ma);
    scanf("%d",&ma2);
    scanf("%d",&na);
    //Выделение памяти
    //Ограничения
    //Матрица ограничений-равенств A1
    double **A = new double*[ma];
    for(int i=0; i<ma; ++i){
        A[i] = new double[na];}
    //Вектор свободных элементов B1
    double **B = new double*[ma];
    for(int i=0; i<ma; ++i){
        B[i] = new double[1];}
    //Матрица ограничений-неравенств A2
    double **A2 = new double*[ma2];
    for(int i=0; i<ma2; ++i){
        A2[i] = new double[na];}
    //Вектор свободных элементов B2
    double **B2 = new double*[ma2];
    for(int i=0; i<ma2; ++i){
        B2[i] = new double[1];}
    //~
    //Целевой функционал
    //Матрица коэффициентов квадратичной части Q
    double **Q = new double*[na];
    for(int i=0; i<na; ++i){
        Q[i] = new double[na];}
    double **QB = new double*[na];
    for(int i=0; i<na; ++i){
        QB[i] = new double[na];}
    //Вектор коэффициентов линейной части P
    double **P = new double*[na];
    for(int i=0; i<na; ++i){
        P[i] = new double[1];}
    double **PB = new double*[1];
    for(int i=0; i<1; ++i){
        PB[i] = new double[na];}
    //~
    //Вспомогательные матрицы
    //Матрица независимых векторов A1
    double **An = new double*[ma];
    for(int i=0; i<ma; ++i){
        An[i] = new double[na];}
    //Матрица независимых векторов A2
    double **A2n = new double*[ma2];
    for(int i=0; i<ma2; ++i){
        A2n[i] = new double[na];}
    //Псевдообратная матрица к An
    double **AnPlus = new double*[na];
    for(int i=0; i<na; ++i){
        AnPlus[i] = new double[ma];}
    double **C = new double*[ma];
    for(int i=0; i<ma; ++i){
        C[i] = new double[na];}
    //Псевдообратная матрица к C
```

```

double **CPlus = new double*[1];
for(int i=0; i<1; ++i){
    CPlus[i] = new double[ma];}
double **APlus = new double*[na];
for(int i=0; i<na; ++i){
    APlus[i] = new double[ma];}
double **up = new double*[ma2];
for(int i=0; i<ma2; ++i){
    up[i] = new double[t+1];}
double **s = new double*[ma2];
for(int i=0; i<ma2; ++i){
    s[i] = new double[1];}
double **sT = new double*[1];
for(int i=0; i<1; ++i){
    sT[i] = new double[ma2];}
double **Xnull = new double*[na];
for(int i=0; i<na; ++i){
    Xnull[i] = new double[1];}
double **XEnd = new double*[na];
for(int i=0; i<na; ++i){
    XEnd[i] = new double[1];}
double **XEndT = new double*[1];
for(int i=0; i<1; ++i){
    XEndT[i] = new double[na];}
double **I1 = new double*[1];
for(int i=0; i<1; ++i){
    I1[i] = new double[na];}
double **I2 = new double*[1];
for(int i=0; i<1; ++i){
    I2[i] = new double[1];}
double **I3 = new double*[1];
for(int i=0; i<1; ++i){
    I3[i] = new double[1];}
//Вектор независимых элементов P
double **Pn = new double*[na];
for(int i=0; i<na; ++i){
    Pn[i] = new double[1];}
//Первая часть линейного функционала
double **P1 = new double*[1];
for(int i=0; i<1; ++i){
    P1[i] = new double[na];}
//Вторая часть линейного функционала
double **P2 = new double*[1];
for(int i=0; i<1; ++i){
    P2[i] = new double[na];}
//~
//Ввод данных
//Квадратичная часть функционала Q
for(int i=0;i<na;i++){
    for(int j=0;j<na;j++){
        scanf("%lf", &Q[i][j]);}}
for(int i=0;i<na;i++){
    for(int j=0;j<na;j++){
        QB[i][j] = Q[i][j];}}
//Линейная часть функционала P
for(int i=0;i<na;i++){
    for(int j=0;j<1;j++){
        scanf("%lf", &P[i][j]);}}
for(int i=0;i<1;i++){
    for(int j=0;j<na;j++){
        PB[i][j] = P[j][i];}}
//Матрица ограничений-равенств A1
for(int i=0;i<ma;i++){
    for(int j=0;j<na;j++){
        scanf("%lf", &A[i][j]);}}

```

```

//Матрица ограничений-неравенств A2
for(int i=0;i<ma2;i++){
    for(int j=0;j<na;j++){
        scanf("%lf", &A2[i][j]);}}
//Вектор свободных элементов B1
for(int i=0;i<ma;i++){
    for(int j=0;j<1;j++){
        scanf("%lf", &B[i][j]);}}
//Вектор свободных элементов B2
for(int i=0;i<ma2;i++){
    for(int j=0;j<1;j++){
        scanf("%lf", &B2[i][j]);}}

//~
//Выводы на экран начальных данных
//Вывод начальной матрицы ограничений-равенств A1
print_matrix (A, ma, na, "A1");
//Вывод начальной матрицы ограничений-неравенств A2
print_matrix (A2, ma2, na, "A2");
//Вывод вектора свободных элементов B1
print_matrix (B, ma, 1, "B1");
//Вывод вектора свободных элементов B2
print_matrix (B2, ma2, 1, "B2");
//Вывод квадратичной части функционала
print_matrix (Q, na, na, "Q");
//Вывод линейного функционала
print_matrix (P, na, 1, "P");
//~
//Тело программы
//Функция поиска столбца с максимальной нормой
find_max_norm (A, ma, na, k, norm);
//Номер итерации
z = 1;
column_swaps (A, A2, ma, na, An, A2n, ma2, z, k, P, Pn);
//Поиск псевдообратной матрицы An

find_pseudo_inverse_first (An, AnPlus, z, ma);
//Рекуррентный процесс
do {
    if (na > z) {
        //Расчёт матрицы C
        C_calc (A, ma, na, z, An, AnPlus, C);
        //Поиск номера столбца с максимальной нормой
        find_max_norm (C, ma, na-z, k, norm);
        if ((norm > tol)) {
            //Итерация +1
            z=z+1;
            //Перестановка столбцов в матрицах A, An, P, Pn, A2 и A2n
            column_swaps (A, A2, ma, na, An, A2n, ma2, z, k, P, Pn);
            //Поиск псевдообратной матрицы к C
            find_pseudo_inverse (C, CPlus, k, ma);
            //Пересчёт псевдообратной матрицы An+ с учётом C+
            refind_pseudoinversion_An (A, ma, na, z, CPlus, AnPlus, k);
            //Добавление к An+ C+
            for(int i=0; i<ma; ++i){
                AnPlus[z-1][i]=CPlus[0][i];}
            //Проверка условия, что матрица ограничений-равенств A1 не пуста
            //~
        }
        check = true;
        for(int i=0; i<ma; ++i){
            for(int j=0; j<na; ++j){
                if (A[i][j] != 0) {
                    check = false;}}}
    }
    if (check == true){

```

```

        break;}
} while (norm > tol);
//Конец рекуррентного процесса
double **B1new = new double*[z];
for(int i=0; i<z; ++i){
    B1new[i] = new double[1];}
//Пересчёт столбца свободных элементов B1
multiplication_matrix(AnPlus, z, ma, B, ma, 1, B1new);
//Проверка условия, что матрица ограничений-равенств A1 не пуста
if (check == true) {
    // пишем z, так как в случае, когда A1 пуста, z = na
    multiplication_matrix(AnPlus, z, ma, B, ma, 1, Xnull);
    cout<<endl<<z<<endl;
    for(int i=0;i<z;i++){
        for(int j=0;j<1;j++){
            XEnd[i][j] = Xnull[i][j];
            XEndT[j][i] = XEnd[i][j];}}
    print_matrix (XEnd, z, 1, "X");
    multiplication_matrix(XEndT, 1, z, QB, z, z, I1);
    multiplication_matrix(I1, 1, z, XEnd, z, 1, I2);
    multiplication_matrix(PB, 1, z, XEnd, z, 1, I3);
    I3[0][0] = I3[0][0] + I2[0][0];
    print_matrix (I3, 1, 1, "I(x)");
    cout<<endl<<endl<<"The solution is found, matrix A1 is empty"<<endl;}
else {
    double **W = new double*[z];
    for(int i=0; i<z; ++i){
        W[i] = new double[na-z];}
    //Расчёт первой части линейного функционала
    functional_conversion_P1 (A, ma, na, z, Pn, P, AnPlus, P1);
    //Разбиения функционала Q на 4 части: luc - left up corner, ruc - right up
corner, ldc - left down corner, rdc - right down corner
    double **Qluc = new double*[z];
    for(int i=0; i<z; ++i){
        Qluc[i] = new double[z];}
    for(int i=0; i<z; ++i){
        for(int j=0; j<z; ++j){
            Qluc[i][j] = Q[i][j];}}

    double **Qruc = new double*[z];
    for(int i=0; i<z; ++i){
        Qruc[i] = new double[na-z];}
    for(int i=0; i<z; ++i){
        for(int j=0; j<na-z; ++j){
            Qruc[i][j] = Q[i][j];}}

    double **Qldc = new double*[na-z];
    for(int i=0; i<na-z; ++i){
        Qldc[i] = new double[z];}
    for(int i=0; i<na-z; ++i){
        for(int j=0; j<z; ++j){
            Qldc[i][j] = Q[i][j];}}

    double **Qrdc = new double*[na-z];
    for(int i=0; i<na-z; ++i){
        Qrdc[i] = new double[na-z];}
    for(int i=0; i<na-z; ++i){
        for(int j=0; j<na-z; ++j){
            Qrdc[i][j] = Q[i][j];}}

    //Расчёт матрицы W
    multiplication_matrix(AnPlus, z, ma, A, ma, na-z, W);
    //Расчёт второй части линейного функционала и пересчёт квадратичной части
функционала
    functional_conversion_P2_and_Q(W, na, z, Qluc, Qruc, Qldc, Qrdc, Q, P2, B1new);
    //Пересчёт линейной части функционала

```



```

for(int j=0; j<na-z; ++j){
    P[j][0] = P1[0][j] + P2[0][j];}
//Вывод на экран пересчитанной линейной части
double **A2T = new double*[na-z];
for(int i=0; i<na-z; ++i){
    A2T[i] = new double[ma2];}
double **E2 = new double*[ma2];
for(int i=0; i<ma2; ++i){
    E2[i] = new double[na-z];}
double **E1 = new double*[ma2];
for(int i=0; i<ma2; ++i){
    E1[i] = new double[1];}
double **h = new double*[ma2];
for(int i=0; i<ma2; ++i){
    h[i] = new double[1];}
double **G = new double*[ma2];
for(int i=0; i<ma2; ++i){
    G[i] = new double[ma2];}
double **U = new double*[na-z];
for(int i=0; i<na-z; ++i){
    U[i] = new double[1];}
double **X = new double*[na-z];
for(int i=0; i<na-z; ++i){
    X[i] = new double[1];}
double **splus = new double*[ma2];
for(int i=0; i<ma2; ++i){
    splus[i] = new double[1];}
double **hplus = new double*[ma2];
for(int i=0; i<ma2; ++i){
    hplus[i] = new double[1];}
double **q = new double*[ma2];
for(int i=0; i<ma2; ++i){
    q[i] = new double[1];}
double **Gplus = new double*[ma2];
for(int i=0; i<ma2; ++i){
    Gplus[i] = new double[ma2];}
inversion(Q, na-z);
multiplication_matrix(A2, ma2, na-z, Q, na-z, na-z, E2);
multiplication_matrix(E2, ma2, na-z, P, na-z, 1, E1);
for(int j = 0; j < ma2; ++j){
    h[j][0] = 0.5*E1[j][0]+B2[j][0];}
for(int i = 0; i < na-z; ++i){
    for(int j = 0; j < ma2; ++j){
        A2T[i][j] = A2[j][i];}}
multiplication_matrix(E2, ma2, na-z, A2T, na-z, ma2, G);
for(int i = 0; i < ma2; ++i){
    for(int j = 0; j < ma2; ++j){
        G[i][j] = 0.25*G[i][j];}}
for(int i = 0; i < ma2; ++i){
    up[i][0] = 0;}
for(int p = 1; p < t+1; ++p){
    qua +=1;
    r = 1;
    for(int i = 0; i < ma2; ++i){
        sum1 = 0;
        sum2 = 0;
        if (i > 0){
            for(int j = 0; j < i; ++j){
                sum1 = sum1 + G[i][j]*up[j][p];}}
            if (i+1 < ma2){
                for(int j = i+1; j < ma2; ++j){
                    sum2 = sum2 + G[i][j]*up[j][p-1];}}
            up[i][p] = -1*(h[i][0]/2 + sum1 + sum2)/G[i][i];
            if (up[i][p] < 0)
                {up[i][p] = 0;}

```

```

        s[i][0] = up[i][p];}
// Проверка условий Куна-Таккера
l = 0;
for(int i = 0; i < ma2; ++i){
    if (s[i][0] > 0){
        hplus[l][0] = h[i][0];
        q[l][0] = i;
        l += 1;}}
for (int i = 0; i < l; ++i){
    f1 = q[i][0];
    for (int j = 0; j < l; ++j){
        f2 = q[j][0];
        Gplus[i][j] = G[f1][f2];}}
// Расчёт splus
inversion (Gplus, l);
multiplication_matrix(Gplus, l, l, hplus, l, l, splus);
for (int i = 0; i < l; ++i){
    splus[i][0] = -0.5*splus[i][0];}
// Расчёт X
multiplication_matrix(A2T, na-z, ma2, s, ma2, 1, U);
for(int j = 0; j < na-z; ++j){
    U[j][0] = U[j][0] + P[j][0];}
multiplication_matrix(Q, na-z, na-z, U, na-z, 1, X);
for(int j = 0; j < na-z; ++j){
    X[j][0] = -0.5*X[j][0];}
//Условие
if (r != 0){
    for(int j = 0; j < l; ++j){
if (splus[l][0] < 0){
        r = 0;}}}}
//Если условия выполнены
if (r == 1){
    //cout<<endl<<endl<<"K-T conditions are completed";
    break;}}

for(int i=0;i<na-z;i++){
    for(int j=0;j<l;j++){
        XEnd[i][j] = X[i][j];}}
for(int i=na-z;i<na;i++){
    for(int j=0;j<l;j++){
        XEnd[i][j] = 0;}}
print_matrix (XEnd, na, 1, "X");
for(int i=0;i<na;i++){
    for(int j=0;j<l;j++){
        XEndT[j][i] = XEnd[i][j];}}
multiplication_matrix(XEndT, 1, na, QB, na, na, I1);
multiplication_matrix(I1, 1, na, XEnd, na, 1, I2);
multiplication_matrix(PB, 1, na, XEnd, na, 1, I3);
I3[0][0] = I3[0][0] + I2[0][0];
print_matrix (I3, 1, 1, "I(x)");
}
cout<<endl<<endl<<"Quantity of iterations: "<< qua;
cout<<endl;
return 0;
}

```