МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение высшего образования
«Тольяттинский государственный университет»

Кафедра «Прикладная математика и информатика»
(наименование)

01.03.02 Прикладная математика и информатика
(код и наименование направления подготовки / специальности)

Компьютерные технологии и математическое моделирование
(направленность (профиль) / специализация)

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА (БАКАЛАВРСКАЯ РАБОТА)

на тему «Математические методы прогнозирования и управления многопродуктовой экономикой на основе модели межотраслевого баланса»

Обучающийся Р. Е. Шаврин
(Инициалы Фамилия) (личная подпись)

Руководитель к.т.н., доцент, Н. А. Сосина
(ученая степень (при наличии), ученое звание (при наличии), Инициалы Фамилия)

Консультант к.п.н., доцент, С. А. Гудкова
(ученая степень (при наличии), ученое звание (при наличии), Инициалы Фамилия)

Тольятти 2024

Аннотация

Тема бакалаврской работы: «Математические методы прогнозирования и управления многопродуктовой экономикой на основе модели межотраслевого баланса».

В ходе выполнения исследований по бакалаврской работе был проведен анализ существующих методов прогнозирования, описаны статическая и динамическая модели межотраслевого баланса.

Бакалаврская работа состоит из введения, трёх разделов, заключения и списка использованной литературы.

Во введении рассматривается актуальность работы, а также задачи и цель ВКР, которая заключается в исследовательской работе в рамках межотраслевого баланса.

В первом разделе проводится сравнительный анализ методов прогнозирования и управления многопродуктовой экономикой, а также рассматривается практическое применение межотраслевого баланса.

Во втором разделе описана математическая модель межотраслевого баланса. Также произведен выбор языка программирования и спроектирована блок-схема для дальнейшей разработки ПО.

В третьем разделе описана реализация ПО. Также произведено функциональное тестирование и вычислительный эксперимент для проверки корректности полученных данных. На основе полученных данных произведен анализ.

В заключении представлены результаты выполнения бакалаврской работы.

Бакалаврская работа состоит из 57 страниц, 16 рисунков, 3 таблицы, 20 формул, 30 источников и 1 листинга.

Abstract

Topic of the bachelor's thesis: “Mathematical methods for forecasting and managing a multi-product economy based on an input-output balance model.”

During the bachelor's thesis research, an analysis of existing forecasting methods was carried out, and static and dynamic models of the interindustry balance were described.

The bachelor's thesis consists of an introduction, three sections, a conclusion and a list of references.

The introduction discusses the relevance of the work, as well as the tasks and purpose of the WRC, which is research work within the framework of intersectoral balance.

The first section provides a comparative analysis of methods for forecasting and managing a multi-product economy, and also examines the practical application of the input-output balance.

The second section describes the mathematical model of the interindustry balance. A programming language was also selected and a block diagram was designed for further software development.

The third section describes the software implementation. Functional testing and a computational experiment were also carried out to verify the correctness of the data obtained. Based on the data obtained, analysis was carried out.

In conclusion, the results of the bachelor's work are presented.

Bachelor's work consists of 57 pages, 16 figures, 3 tables, 20 formulas, 30 sources and 1 listing.

Содержание

Введение

Современная экономика, особенно в условиях глобализации и развития технологий, требует высокой точности и эффективности прогнозирования и управления. «Важно отметить, что в многопродуктовой экономике отношения между различными секторами играют ключевую роль в формировании структуры и динамики рынка. математические методы прогнозирования и управления на основе моделей межотраслевого баланса представляют собой мощный инструмент анализа и принятия решений в таких условиях» [1].

Актуальность данной темы обусловлена не только сложностью современной экономической системы, но и необходимостью оперативного реагирования на изменения внешних и внутренних факторов. «Глубокое понимание взаимосвязей между отраслями и способность прогнозировать их развитие являются ключевыми составляющими успешного функционирования как отдельных предприятий, так и всей экономики страны» [2].

В настоящее время исследования в области математических методов прогнозирования и управления многопродуктовой экономикой на основе модели межотраслевого баланса находятся на пике интереса. Необходимость постоянного обновления и усовершенствования моделей межотраслевого баланса требует значительных инвестиций в научные исследования и развитие технологий, что является стратегическим приоритетом для многих стран и компаний. Улучшение качества данных и разработка более гибких и адаптируемых моделей являются ключевыми направлениями для преодоления существующих проблем в прогнозировании и управлении экономикой. «С развитием вычислительной техники и методов анализа данных появляются новые возможности для улучшения точности и оперативности прогнозирования. Исследователи активно работают над разработкой новых моделей и алгоритмов, а также углублением понимания взаимосвязей между отраслями и факторами, влияющими на экономические процессы» [3].

Однако, несмотря на широкое использование, существует ряд нерешенных проблем в области математических методов прогнозирования и управления многопродуктовой экономикой на основе модели межотраслевого баланса. Одной из таких проблем является несовершенство моделей и методов анализа, что может привести к неточным прогнозам и неправильным решениям. «Также существует проблема нехватки данных или их недостаточной качественности для построения достоверных моделей, особенно в случае новых или быстро развивающихся отраслей. Кроме того, в условиях быстрой изменчивости современной экономики существует проблема адаптации моделей к новым условиям и факторам, которые могут значительно влиять на деятельность компаний и экономические процессы в целом» [4].

Объектом исследования бакалаврской работы является модель межотраслевого баланса

Предмет исследования – базы данных, алгоритмы, программные решения, библиотеки, которые будут использованы для построения модели межотраслевого баланса.

Цель работы – разработка алгоритма на основе модели межотраслевого баланса и создание программного обеспечения, которое будет выводить схему межотраслевого баланса.

Задачи выпускной квалификационной работы включают в себя:

– изучение актуальных исследований и литературы в области математических методов прогнозирования и управления экономикой на основе модели межотраслевого баланса;

- описание математических моделей межотраслевого баланса;
- описание блок-схемы для решения задачи межотраслевого баланса;
- выбор языка программирования и среды для разработки алгоритма решения задачи межотраслевого баланса;
- реализация программного обеспечения и проведение функционального тестирования;

– анализ и прогнозирование на основе полученных результатов и проведение вычислительного эксперимента для проверки корректности полученных данных.

Данная работа включает в себя введение, три главы, заключение и список литературы.

В первом разделе выполнена постановка задачи, а также осуществляется выбор языка программирования и среды для разработки ПО, рассматривается практическое применение межотраслевого баланса.

Во втором разделе, описаны статическая и динамическая модели межотраслевого баланса, спроектирована блок-схема алгоритма решения поставленной задачи.

В третьем разделе, описан процесс реализации программного обеспечения и произведено тестирование ПО, произведены анализ и прогнозирование на основе полученных данных, выполнен вычислительный эксперимент с целью проверки полученных данных.

1 Описание математических методов прогнозирования и управления многопродуктовой экономикой

1.1 Практическое применение модели межотраслевого баланса

«Математические методы прогнозирования и управления многопродуктовой экономикой на основе модели межотраслевого баланса имеют широкое применение в различных отраслях и секторах экономики. В частности, они играют ключевую роль в стратегическом планировании и оперативном управлении отдельными компаниями и предприятиями» [5].

Применение этих методов для отдельных кампаний заключается в использовании модели межотраслевого баланса для анализа внешних и внутренних факторов, влияющих на их деятельность. «Компании могут использовать такие модели для прогнозирования спроса на свою продукцию, определения оптимальных стратегий производства и распределения ресурсов, а также для оценки возможных рисков и разработки мер по их смягчению. Например, производитель автомобилей может использовать модель межотраслевого баланса для прогнозирования спроса на автомобили в зависимости от изменений в макроэкономических показателях, таких как уровень безработицы или рост ВВП, и адаптировать свою производственную стратегию соответственно» [6]. Также эти методы могут применяться для следующих параметров:

- для разработки экономических планов на уровне государства или отдельных регионов. Эти планы могут включать в себя прогнозирование производства, потребления, инвестиций и экспорта в различных отраслях экономики;
- для оценки экономические эффекты различных событий или политических мероприятий, таких как изменение налогов, торговые соглашения или инфраструктурные проекты;

– для стратегического планирования и принятия решений о развитии бизнеса, так как на основе анализа межотраслевых связей и взаимосвязей могут быть выявлены новые возможности для роста и определены оптимальные стратегии развития;
– для оценки потенциальных инвестиционных проектов и их влияния на экономику, это помогает инвесторам принимать обоснованные решения о распределении капитала и выборе проектов для инвестирования;
– для оценки экономических рисков и разработки стратегий управления ими, так как основе анализа взаимосвязей между различными отраслями и секторами экономики можно определить потенциальные уязвимости и разработать меры по их смягчению.

В рамках поставленной задачи я буду производить исследовательскую работу в рамках государственной макроэкономики и построение модели межотраслевого баланса будет на основе взаимодействия таких сфер, как: сельское хозяйство, металлургия, химическая промышленность, машиностроение и энергетика. На основе этих сфер будет наиболее наглядно понятен принцип построения модели межотраслевого баланса.

1.2 Обзор методов прогнозирования и управления многопродуктовой экономикой

В настоящий момент существует множество методов прогнозирования и управления многопродуктовой экономикой. Передо мной была поставлена задача построить модель межотраслевого баланса для расчёта затрат на открытие трёх новых офисов кампании и посчитать прогноз окупаемости вложенных средств. Для этой задачи не все методы будут оптимальны, поэтому произведем сравнительный анализ методов прогнозирования и управления многопродуктовой экономикой (таблица 1).

Таблица 1 – Сравнительный анализ методов прогнозирования и управления многопродуктовой экономикой

Метод прогнозирования	Преимущества метода	Недостатки метода
Методы линейного программирования	Эффективны для оптимизации распределения ресурсов в условиях многопродуктовой экономики; Обладают высокой точностью и возможностью учета различных ограничений.	Ограничены линейными отношениями между переменными, что может привести к упрощенным моделям, не учитывающим сложные взаимосвязи в экономике.
Векторная авторегрессия	Позволяет учитывать динамику временных рядов и взаимосвязи между ними; Может быть применена для прогнозирования различных параметров экономики на основе исторических данных.	требует достаточного объема данных для точного прогнозирования; могут возникать проблемы с интерпретацией результатов в случае большого числа переменных.
Методы многомерного статистического анализа	Позволяют выявить структурные зависимости в данных и выделить основные факторы, влияющие на экономику; Могут быть полезны для классификации отраслей и выявления паттернов.	Могут быть менее точными в сравнении с другими методами прогнозирования, особенно в условиях быстрой изменчивости экономических процессов
Методы искусственного интеллекта и машинного обучения	Позволяют работать с большими объемами данных и выявлять сложные нелинейные взаимосвязи; Могут обеспечить высокую точность прогнозирования в условиях переменчивости экономических процессов.	Требуют большего объема вычислительных ресурсов и специалистов с навыками работы с алгоритмами машинного обучения; Могут быть менее интерпретируемыми в сравнении с классическими статистическими методами.

Исходя из сравнительного анализа в таблице 2, можно сделать вывод, что наиболее подходящими для поставленной задачи в рамках анализа макроэкономических показателей будет анализ и прогнозирование на основе полученных данных, которое производится пользователем разработанного программного обеспечения.

2 Описание математической модели межотраслевого баланса и архитектуры системы

2.1 Статическая математическая модель межотраслевого баланса

Модель межотраслевого баланса бывает двух видов: статическая и динамическая. «Для того, чтобы наиболее эффективно и наглядно построить модель межотраслевого баланса необходимо рассмотреть два типа модели и на основе этого необходимо выбрать наиболее подходящую модель. Начнём со статической модели» [7].

Эффективное функционирование экономики предполагает наличие баланса между отдельными отраслями. «Каждая отрасль при этом выступает с одной стороны, как производитель некоторой продукции, а с другой – как потребитель продуктов, вырабатываемых другими отраслями. Для наглядного выражения взаимной связи между отраслями используют таблицы определенного вида, которые называют таблицами межотраслевого баланса» [8].

«Межотраслевой баланс – это таблица статистических данных, в которой отражено производство продукции, и ее распределение между отраслями. С помощью модели межотраслевого баланса можно выполнять плановые расчеты» [9].

«Межотраслевые балансы могут разрабатываться на плановый и отчетный период в натуральном, натурально-стоимостном и стоимостном выражении» [10].

Межотраслевой баланс может быть представлен в виде схемы и модели. Схема межотраслевого баланса производства и распределения общественного продукта в стоимостном выражении приведена ниже (таблица 2).

Таблица 2 – Схема межотраслевого баланса производства и распределения общественного продукта в стоимостном выражении

Отрасли-Производители	Отрасли-потребители				Промежуточное потребление	Конечное потребление	Валовый выпуск
	1	2	…	n			
1	x_{11}	x_{12}	…	x_{1n}	$\sum_{j=1}^{n} x_{1j}$	y_1	x_1
2	x_{21}	x_{22}	…	x_{2n}	$\sum_{j=1}^{n} x_{2j}$	y_2	x_2
…	…	…	…	…	…	…	…
n	x_{n1}	x_{n2}	…	x_{nn}	$\sum_{j=1}^{n} x_{nj}$	y_n	x_n
Промежуточные затраты	$\sum_{i=1}^{n} x_{i1}$	$\sum_{i=1}^{n} x_{i2}$	…	$\sum_{i=1}^{n} x_{i}$	$\sum_{i=1}^{n}\sum_{j=1}^{n} x_{ij}$	$\sum_{i=1}^{n} y_i$	$\sum_{i=1}^{n} x_i$
Валовая добавленная стоимость	v_1	v_2	…	v_n	$\sum_{j=1}^{n} v_j$		
Валовый выпуск	x_1	x_2	…	x_n	$\sum_{j=1}^{n} x_j$		

Все народное хозяйство представляется в виде совокупности n отраслей. Вся продукция отраслей разделена на промежуточную и конечную.

На схеме использованы обозначения:

x_{ij}– затраты продукции отрасли i на производство продукции отрасли j;

y_i –конечная продукция отрасли i;

x_i - валовая продукция i-й отрасли;

v_i – добавленная стоимость j-ой отрасли.

В схеме МОБ можно выделить три раздела или квадранта.

«I раздел представляет собой матрицу элементов, стоящих на пересечении п первых строк и п первых столбцов баланса. Этот раздел отражает межотраслевые взаимосвязи по использованию продукции на текущее производственное (промежуточное) потребление» [11].

«Величины $\sum_{j=1}^{n} x_{ij}$ (i= $\overline{1,n}$) характеризуют производственное потребление продукции i-ой отрасли, величины $\sum_{j=1}^{n} x_{ij}$ (j= $\overline{1,n}$) суммы производственных затрат j-ой отрасли. Число $\sum_{i=1}^{n}\sum_{j=1}^{n}x_{ij}$ равно сумме всех производственных затрат всех отраслей. Это так называемый промежуточный продукт народного хозяйства» [12].

«II раздел расположен справа от столбца промежуточного потребления. Этот раздел дан укрупненно, в виде одного столбца величин y_i. В развернутой схеме отображается использование на личное и общественное потребление, валовое накопление. Кроме того, в конечный продукт входит сальдо экспорта-импорта продукции. II раздел отражает отраслевую и материально-вещественную структуру конечного использования общественного продукта» [13].

«III раздел расположен под первым. Раздел также дан укрупненно, в виде строки величин v_j . В развернутой схеме отражаются элементы добавленной стоимости: потребление основного капитала, прибыль, заработная плата; косвенные налоги, субсидии. III раздел отражает стоимостную структуру валового внутреннего продукта» [10].

«В схеме МОБ совмещаются два частных межотраслевых баланса – баланс распределения продукции (I и II раздел) и баланс затрат (I и III раздел)» [14].

Однако для прогнозных расчетов соотношения в виде (1) и (2) малопригодны в силу сложности определения элементов промежуточного потребления. Поэтому определение последних осуществляют на базе нормативов, называемых коэффициентами прямых затрат, или технологическими коэффициентами, и выражается по формуле (формула 1).

$$a_{ij} = \frac{x_{ij}}{x_j} \quad . \tag{1}$$

«Коэффициенты прямых затрат a_{ij} показывают, какое количество продукции i-ой отрасли необходимо для производства единицы валовой продукции j-ой отрасли. В совокупности они образуют матрицу прямых затрат» [15]. При этом предположении величины межотраслевых потоков могут быть определены по формуле (формула 2)

$$x_{ij} = a_{ij} * x_j \ . \tag{2}$$

Запишем систему уравнений с учётом формулы 2 (формула 3).

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n + y_1 = x_1; \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n + y_2 = x_2; \\ \ldots \\ a_{n1}x_1 + a_{n2}x_2 + \ldots + a_{nn}x_n + y_n = x_n. \end{cases} \tag{3}$$

Обозначим через X вектор валового выпуска (формула 4), а через Y вектор конечной продукции (формула 5).

$$X = \begin{bmatrix} x_1 \\ x_2 \\ \ldots \\ x_n \end{bmatrix} , \tag{4}$$

$$Y = \begin{bmatrix} y_1 \\ y_2 \\ \ldots \\ y_n \end{bmatrix} . \tag{5}$$

Запишем систему (формула 3) в матричной форме (формула 6).

A*X + Y = X или (E - A)*X = Y . (6)

Единичная матрица E размерности n * n представлена ниже (формула 7).

$$E = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 \end{bmatrix} . \qquad (7)$$

«Система уравнений (формула 3), записанная как векторное уравнение (формула 6) называется уравнением Леонтьева. С его помощью выполняют планирование и прогнозирование в экономике» [16].

Для расчёта матрицы полных затрат выразим X из балансового соотношения формулы системы уравнений в матричной форме (формула 8).

$$X = (E - A)^{-1} * Y, \qquad (8)$$

где $(E - A)^{-1}$ – матрица, обратная $(E - A)$.

Она также называется матрицей коэффициентов полных затрат.

Коэффициенты полных затрат b_{ij} показывают, какое количество продукции i-ой отрасли необходимо, для получения единицы конечной продукции j-ой отрасли. Экономический смысл коэффициентов полных затрат b_{ij} будет понятен, если переписать соотношения из формулы 8 в развернутом виде (формула 9)

$$\begin{cases} b_{11}x_1 + b_{12}x_2 + \dots + b_{1n}y_n = x_1 \\ b_{21}x_1 + b_{22}x_2 + \dots + b_{2n}y_n = x_2 \\ \dots \\ b_{n1}x_1 + b_{n2}x_2 + \dots + b_{nn}y_n = x_n \end{cases} . \qquad (9)$$

«Положим, что осуществляется выпуск конечной продукции лишь одной (пусть первой) отрасли в размере 1 денежной единицы. Из соотношения (8) очевидно, что для того, чтобы обеспечить конечную продукцию в указанном объеме необходимо обеспечить валовой выпуск продукции отраслей соответственно в объеме $b_{11}, b_{21}, \dots b_{n1}$. Таким образом, элементы первого столбца матрицы B показывают количество валовой продукции отраслей, необходимых для производства единицы конечной продукции первой отрасли. Аналогично можно показать, что элементы j-го столбца матрицы B показывают количество валовой продукции отраслей, необходимых для производства единицы конечной продукции j-й отрасли. Матрица $(E - A)^{-1}$ существует только в том случае, если матрица A является продуктивной, т.е. сумма элементов каждого столбца матрицы коэффициентов прямых затрат меньше 1» [17].

На основе математической модели можно сделать вывод, что статическая модель межотраслевого баланса отлично подходит для решения задач в макроэкономике. Перейдем к описанию динамической модели межотраслевого баланса.

2.2 Динамическая модель межотраслевого баланса

«Рассмотренные выше межотраслевые балансовые модели являются статическими, т. е. такими, в которых все зависимости отнесены к одному моменту времени. Эти модели могут разрабатываться лишь для отдельно взятых периодов, причем в рамках данных моделей не устанавливается связь с предшествующими или последующими периодами. Народнохозяйственная динамика отображается, таким образом, рядом независимо рассчитанных моделей, что очевидно вносит определенные упрощения и сужает возможности анализа» [18].

«К числу таких упрощений прежде всего следует отнести то, что в статических межотраслевых моделях не анализируются распределение, использование и производственная эффективность капитальных вложений. Капиталовложения вынесены из сферы производства в сферу конечного использования вместе с предметами потребления и непроизводственными затратами, т.е. включены в конечный продукт» [19].

«В отличие от статических динамические модели призваны отразить не состояние, а процесс развития экономики, установить непосредственную взаимосвязь между предыдущими и последующими этапами развития и тем самым приблизить анализ на основе экономико-математической модели к реальным условиям развития экономической системы» [20].

«В рассматриваемой здесь динамической модели, являющейся развитием статической межотраслевой модели, производственные капитальные вложения выделяются из состава конечной продукции, исследуются их структура и влияние на рост объема производства. В основе построения модели в виде динамической системы уравнений лежит математическая зависимость между величиной капитальных вложений и приростом продукции. Решение системы, как и в случае статической модели, приводит к определению уровней производства, но в динамическом варианте в отличие от статического эти искомые уровни зависят от объемов производства в предшествующих периодах» [21]. Принципиальная схема первых двух квадрантов динамического межотраслевого баланса приведена в таблице 3.

Таблица 3 – Схема динамического баланса

<table>
<tr><td rowspan="3">Производящие отрасли</td><td colspan="10">Потребляющие отрасли</td></tr>
<tr><td colspan="4">Межотраслевые потоки текущих затрат</td><td colspan="4">Межотраслевые потоки капитальных вложений</td><td rowspan="2">Конечный продукт</td><td rowspan="2">Валовый продукт</td></tr>
<tr><td>1</td><td>2</td><td>…</td><td>n</td><td>1</td><td>2</td><td>…</td><td>n</td></tr>
<tr><td>1</td><td>x_{11}</td><td>x_{12}</td><td>…</td><td>x_{1n}</td><td>$\Delta\Phi_{11}$</td><td>$\Delta\Phi_{12}$</td><td>…</td><td>$\Delta\Phi_{1n}$</td><td>Y_1</td><td>X_1</td></tr>
<tr><td>2</td><td>x_{21}</td><td>x_{22}</td><td>…</td><td>x_{2n}</td><td>$\Delta\Phi_{21}$</td><td>$\Delta\Phi_{22}$</td><td>…</td><td>$\Delta\Phi_{2n}$</td><td>Y_2</td><td>X_2</td></tr>
<tr><td>…</td><td>…</td><td>…</td><td>…</td><td>…</td><td>…</td><td>…</td><td>…</td><td>…</td><td>…</td><td>…</td></tr>
<tr><td>n</td><td>x_{n1}</td><td>x_{n2}</td><td>…</td><td>x_{nn}</td><td>$\Delta\Phi_{n1}$</td><td>$\Delta\Phi_{n2}$</td><td>…</td><td>$\Delta\Phi_{nn}$</td><td>Y_n</td><td>X_n</td></tr>
</table>

«В рассмотренной динамической модели межотраслевого баланса предполагается, что прирост продукции текущего периода обусловлен капиталовложениями, произведенными в этом же периоде. Для сравнительно коротких периодов это предположение может оказаться нереальным, так как существуют известные, иногда довольно значительные отставания во времени (так называемые временные лаги) между вложением средств в производственные фонды и приростом выпуска продукции. Модели, так или иначе учитывающие лаг капитальных вложений, образуют особую группу динамических моделей межотраслевого баланса» [22].

«Из теоретических моделей данного типа следует назвать прежде всего линейную динамическую межотраслевую модель Леоньтева, в которой капитальные вложения представлены в виде так называемого инвестиционного блока в форме Леонтьева. Математическим обобщением этой и ряда других динамических моделей является динамическая модель в матричной форме Неймана, основанная на математической теории равномерного пропорционального роста экономики (так называемая магистральная теория)» [23].

«Модель содержит две матрицы межотраслевых потоков. Матрица текущих производственных затрат с элементами x_{ij} совпадает с соответствующей матрицей статического баланса. Элементы второй матрицы $\Delta\Phi_{ij}$ показывают, какое количество продукции i-й отрасли направлено в текущем периоде в j-ю отрасль в качестве производственных капитальных вложений в ее основные фонды» [24].

Материально это выражается в приросте в потребляющих отраслях производственного оборудования, сооружений, производственных площадей, транспортных средств и др.

«В статическом балансе потоки капиталовложений не дифференцируются по отраслям-потребителям и отражаются общей величиной в составе конечной продукции Y_i каждой i-й отрасли. В динамической схеме конечный продукт Y_i включает продукцию i-й отрасли, идущую в личное и общественное потребление, накопление непроизводственной сферы, прирост оборотных фондов, незавершенного строительства, на экспорт» [25].

Таким образом, сумма потоков капиталовложений и конечного продукта динамической модели равна конечной продукции статического баланса и выражается по формуле ниже (формула 10).

$$\sum_{j=1}^{n} \Delta\Phi_{ij} + Y_i' = Y_i. \qquad (10)$$

Исходя из формулы 10 уравнение распределения продукции в динамическом балансе преобразуется в следующее уравнение (формула 11).

$$X_i = \sum_{j=1}^{n} x_{ij} + \sum_{j=1}^{n} \Delta\Phi_{ij} + Y_i' ;\ \mathrm{i} = \overline{1,n}\,. \qquad (11)$$

Межотраслевые потоки текущих затрат можно выразить, как в статической модели, через валовую продукцию отраслей с помощью коэффициентов прямых материальных затрат

В отличие от потоков текущих затрат межотраслевые потоки капитальных вложений связаны не со всей величиной выпуска продукции, а обусловливают прирост продукции.

Причем в рассматриваемой модели предполагается, что прирост продукции текущего периода обусловлен вложениями, произведенными в этом же периоде. Если текущий период обозначить через t, то прирост продукции ΔX_j равен разности абсолютных уровней производства в период t и в предшествующий (t - 1)-й период и выражается по формуле ниже (формула 12).

$$\Delta X_j = X_j^{(t)} - X_j^{(t-1)}\ . \qquad (12)$$

Полагая, что прирост продукции пропорционален приросту производственных фондов, можно записать следующее соотношение (формула 13).

$$\Delta\Phi_{ij} = \varphi * \Delta X_j;\ i,j = \overline{1,n}\,. \qquad (13)$$

«Экономический смысл этих коэффициентов заключается в том, что они показывают, какое количество продукции i-й отрасли должно быть вложено в j-ю отрасль для увеличения производственной мощности j-й отрасли на единицу продукции. Предполагается, что производственные мощности используются полностью и прирост продукции равен приросту мощности. Коэффициенты φ_{ij} называются коэффициентами вложений или коэффициентами приростной фондоемкости» [26].

С помощью коэффициентов прямых материальных затрат и коэффициентов вложений φ_{ij} формулу 11 можно представить в следующем виде (формула 14).

$$X_i = \sum_{j=1}^{n} a_{ij} * x_j + \sum_{j=1}^{n} \varphi_{ij} * \Delta X_j + Y_i' ; \; \mathrm{i} = \overline{1,n} \quad . \qquad (14)$$

Формула 14 представляет собой систему линейных разностных уравнений первого порядка. Ее можно привести к обычной системе линейных уравнений, если учесть, что все объемы валовой и конечной продукции относятся к некоторому периоду t, а прирост валовой продукции определен в сравнении с (t - 1)-м периодом (формула 15).

$$X_i^{(t)} = \sum_{j=1}^{n} (a_{ij} + \varphi_{ij}) * X_j^{(t)} + \sum_{j=1}^{n} \varphi_{ij} * X_j^{(t-1)} + Y_i' ; \; \mathrm{i} = \overline{1,n} \; . \qquad (15)$$

«Пусть нам известны уровни валовой продукции всех отраслей в предыдущем периоде (величины $X_j^{(t-1)}$) и конечный продукт отраслей в t-м периоде. Тогда очевидно, что соотношения из формулы 15 представляют собой систему n линейных уравнений с n неизвестными уровнями производства t-ого периода» [27].

Таким образом, решение динамической системы линейных уравнений позволяет определить выпуск продукции в последующем периоде в зависимости от уровня, достигнутого в предыдущем периоде.

Связь между периодами устанавливается через коэффициенты вложений φ_{ij}, характеризующие фондоемкость единицы прироста продукции. Переходя от дискретного анализа к непрерывному, вместо формулы 11 будем окончательно для случая непрерывных изменений иметь следующее выражение (формула 16).

$$X_i = \sum_{j=1}^{n} a_{ij} * X_j + \sum_{j=1}^{n} \varphi_{ij} * \frac{dX_j}{dt} + Y_i' \text{ ; i } = \overline{1,n} \ . \qquad (16)$$

Соотношения по формуле 16 представляют собой систему n линейных дифференциальных уравнений первого порядка с постоянными коэффициентами. Для ее решения помимо матриц коэффициентов прямых материальных текущих затрат и коэффициентов капитальных затрат (вложений) необходимо знать уровни валового выпуска в начальный момент времени t = 0 и закон изменения величины конечного продукта, т.е. вид функций $Y_t'(t)$.

«На основе этих данных путем решения получившейся задачи Коши для системы дифференциальных уравнений из формулы 16 можно найти уровни валового выпуска теоретически для любого момента времени. Практически же более или менее достоверное описание валовых и конечных выпусков как функций времени может быть получено лишь для относительно небольших промежутков времени» [28].

«В динамической модели особую роль играют коэффициенты приростной фондоемкости φ_{ij}. Они образуют квадратную матрицу n-го порядка по формуле ниже (формула 17)» [29].

$$(\varphi_{ij}) = \begin{pmatrix} \varphi_{11} & \varphi_{12} & \cdots & \varphi_{1n} \\ \varphi_{21} & \varphi_{22} & \cdots & \varphi_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ \varphi_{n1} & \varphi_{n1} & \cdots & \varphi_{nn} \end{pmatrix} . \qquad (17)$$

«Каждый столбец в матрице по формуле 17 характеризует для соответствующей j-й отрасли величину и структуру фондов, необходимых для увеличения на единицу ее производственной мощности (выпуска продукции). Матрица коэффициентов приростной фондоемкости дает значительный материал для экономического анализа и планирования капитальных вложений» [30].

В рамках исследовательской работы было принято решение описать на языке программирования Python статическую модель межотраслевого баланса, потому что она наиболее эффективна в рамках анализа отношений макроэкономических показателей.

2.3 Постановка задачи и выбор языка программирования для разработки алгоритма

Перед началом разработки алгоритма необходимо правильно выбрать язык программирования и платформу, которая работает на выбранном языке, чтобы обеспечить точное и оптимальное решение поставленной задачи. Для применения математических методов прогнозирования и управления многопродуктовой экономикой на основе модели межотраслевого баланса потребуется язык программирования, который обладает следующими характеристиками:

- язык должен обеспечивать возможность эффективного выполнения сложных математических вычислений;
- поскольку модели межотраслевого баланса часто требуют обработки больших объемов данных и многократного повторения расчетов, язык программирования должен обладать высокой производительностью выполнения кода;
- желательно, чтобы язык имел богатые библиотеки для вычислений, которые облегчают работу с данными и выполнение сложных математических операций;

– важно иметь возможность визуализации результатов моделирования для анализа данных и принятия решений;
– учитывая сложность и масштаб задачи, язык программирования должен обеспечивать удобство масштабирования и оптимизации кода для работы с различными объемами данных;
– желательно, чтобы язык программирования предлагал интуитивно понятный синтаксис и простой способ организации кода для удобства разработки и поддержки моделей;
– учитывая возможную необходимость в параллельных вычислениях для ускорения процесса моделирования, язык должен обеспечивать поддержку многопоточности или распараллеливания вычислений.

Для того, чтобы определить, какой язык программирования отвечает данным параметрам, рассмотрим сравнительный анализ языков программирования и выявим наиболее подходящий (таблица 4).

Таблица 4 – Сравнительный анализ языков программирования для решения поставленной задачи

Язык программирования	Преимущества метода	Недостатки метода
Python	Имеет обширное сообщество и большое количество библиотек для научных вычислений; Прост в изучении и использовании, имеет чистый и интуитивно понятный синтаксис; Поддержка графической визуализации данных;	Может быть менее производительным по сравнению с некоторыми другими языками при выполнении сложных вычислений из-за интерпретируемости; Некоторые библиотеки могут быть менее эффективными, чем их аналоги в других языках.
R	Специализируется на статистическом анализе данных и имеет множество пакетов для эконометрики и статистического моделирования; Широко используется в научном и академическом сообществе, особенно в области экономики и финансов; Обладает хорошей поддержкой графической визуализации данных.	Может быть не таким гибким и мощным для общих целей, как Python или MATLAB, особенно для выполнения сложных математических операций; Интерфейс может казаться менее интуитивным для новичков в программировании.

Продолжение таблицы 4

Julia	Создавался с целью комбинировать высокую производительность с простотой и удобством использования, особенно для научных вычислений; Обладает высокой производительностью, что делает его привлекательным для вычислительно интенсивных задач; Имеет средства для параллельных вычислений, что может быть полезно для распараллеливания вычислений в моделях экономики.	Пока не так широко распространен и не имеет такого обширного сообщества и библиотек, как Python или MATLAB; Может быть немного сложнее в изучении для тех, кто уже знаком с другими языками программирования.
MATLAB	Отлично подходит для выполнения математических вычислений и реализации алгоритмов; Имеет множество специализированных инструментов и пакетов для решения задач численного анализа и моделирования.	Проприетарный и платный, что может быть проблемой для студентов и исследователей с ограниченным бюджетом; Не так широко распространен во всех областях, как Python или R.

На основе сравнительного анализа можно сделать вывод, что наиболее предпочтительным является язык программирования Python, в виду его широкого функционала для математических вычислений, а также для анализа и визуализации данных. В качестве платформы разработки алгоритма был выбран PyCharm, которые включает в себя весь функционал, необходимый для работы с Python.

2.4 Описание архитектуры системы и системы для разработки программного кода

Для разработки алгоритма, который будет использовать математические методы прогнозирования и управления многопродуктовой экономикой на основе модели межотраслевого баланса необходимо описать систему по компонентам, а также спроектировать систему для дальнейшей разработки программного обеспечения на основе блок-схемы программного обеспечения. Начнём с описания системы по компонентам.

Компонент сбора данных будет использовать файл в формате xlsx. На вход будут подаваться файл, в котором будут матрица коэффициентов прямых затрат и вектор конечной продукции для дальнейших расчётов.

Компонент обработки обрабатывает и анализирует полученные данные. Эта информация используется для построения статической модели межотраслевого баланса.

Компонент визуализации выводит схему межотраслевого баланса производства и распределения общественного продукта в стоимостном выражении. Также будет выведена кнопка экспорта полученного файла в формате xlsx.

На основе описания системы мы можем описать блок-схему для дальнейшей разработки программного обеспечения. Программный код будет написан на Python в системе PyCharm. Основной задачей программного продукта станет расчёт макроэкономических показателей в масштабах государства с применением статической модели межотраслевого баланса (рисунок 2).

В начале работы программного кода будет происходить загрузка матрицы коэффициентов прямых затрат из Excel-файла. После чего будет происходить проверка матрицы на соответствующую размерность в пять строк и столбцов. Если проверка не пройдет, то будет происходить выход из программы с выводом соответствующего сообщения. Если проверка прошла, то далее будет происходить загрузка вектора конечной продукции из Excel-файла и вычисление новой матрицы как разность единичной и матрицы коэффициентов, а также транспонирование полученной матрицы (E - A).

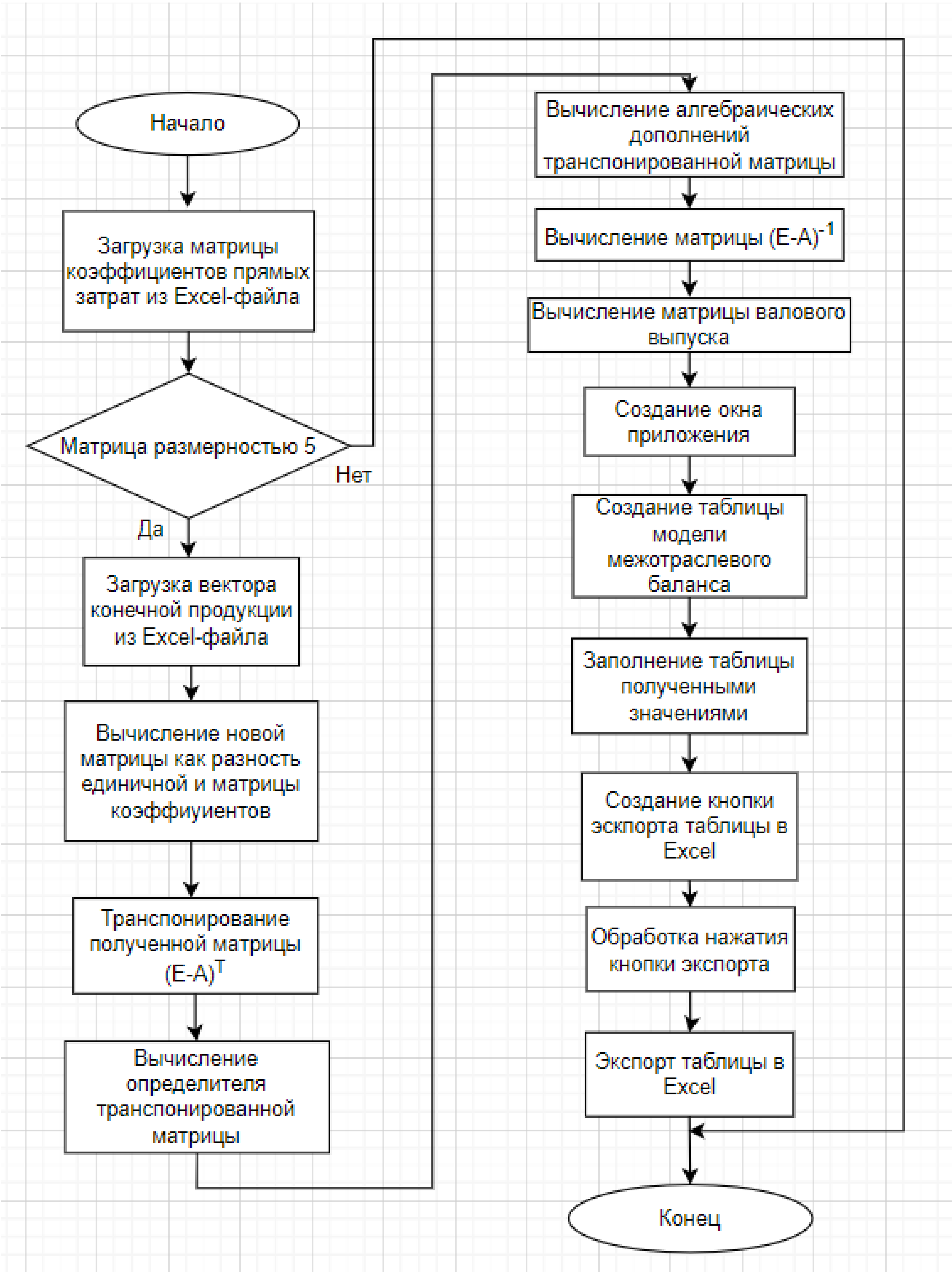

Рисунок 2 – Блок-схема системы для дальнейшей разработки

Из блок-схемы на рисунке 2 мы видим, что после транспонирования матрицы (E-A) будет происходить вычисление определителя транспонированной матрицы и вычисление алгебраических дополнений транспонированной матрицы. Затем будет вычисление обратной матрицы $(E - A)^{-1}$ и вычисление матрицы валового выпуска.

Затем будет запускаться компонент визуализации, в котором будет происходить создание окна приложения, создание таблицы модели межотраслевого баланса и заполнение таблицы полученными вычислениями в ходе работы.

В завершающей части будет создание кнопки экспорта таблицы в формате xlsx для пользователей, а также обработка нажатия кнопки экспорта и экспорт таблицы в Excel.

Выводы по разделу 2:

Во втором разделе ВКР была описана математическая модель статической модели межотраслевого баланса. Также была описана динамическая модель и был осуществлен выбор в пользу статической модели для решения поставленной задачи.

Также был выбран язык программирования и среда для разработки, была описана система по компонентам ввода, обработки и визуализации данных, а также была составлена блок-схема системы для дальнейшей разработки программного кода.

3 Реализация и тестирование программного обеспечения

3.1 Реализация программного обеспечения

Реализация программного обеспечения начинается с импортирования всех необходимых библиотек. В нашем случае это следующие библиотеки:

- Sys, модуль системных операций Python;
- PyQt5, использующийся для создания графического интерфейса пользователя;
- Pandas, использующийся для работы с данными, особенно для чтения и записи данных в различных форматах, таких как Excel;
- Numpy, использующийся для научных вычислений, в данном случае используется для работы с матрицами.

Затем функция pd.read_excel из библиотеки pandas используется для чтения данных из Excel-файла. Она считывает данные из файла 'matrix.xlsx'. В первом случае считываются только первые 5 строк и столбцы A-E, а во втором - столбец G. После этого, проверяем, что размер первой матрицы 5x5. Если условие не соблюдается то, возникает исключение ValueError.

После создается новая матрица new_matrix, инициализированная как единичная матрица (матрица с единицами на главной диагонали и нулями в остальных местах). Далее описаны два цикла:

- первый цикл вычисляет новые значения для элементов на главной диагонали новой матрицы;
- второй цикл вычисляет значения для остальных элементов новой матрицы, меняя знак исходных элементов первой матрицы.

Вложенные циклы проходят по каждому элементу над главной диагональю и меняют его местами с соответствующим элементом под главной диагональю, что и производит транспонирование. Программный код данного блока представлен ниже (рисунок 3).

```python
import sys
from PyQt5.QtWidgets import (QApplication, QFileDialog, QPushButton, QHeaderView, QMainWindow,
                             QTableWidget, QTableWidgetItem, QVBoxLayout, QWidget)
import pandas as pd
import numpy as np

# Загружаем первую матрицу из Excel файла
matrix = pd.read_excel('matrix.xlsx', header=None, nrows=5, usecols="A:E").values

# Загружаем вторую матрицу из Excel файла
matrix2 = pd.read_excel('matrix.xlsx', header=None, usecols="G:G").values

# Проверяем, что матрица 3x3
if matrix.shape != (5, 5):
    raise ValueError("Матрица должна быть размером 5x5")

new_matrix = np.eye(5)
for i in range(5):
    new_matrix[i, i] = 1 - matrix[i, i]

# Умножаем остальные элементы на -1
for i in range(5):
    for j in range(5):
        if i != j:
            new_matrix[i, j] = -matrix[i, j]

#Транспонирование матрицы
for i in range(5):
    for j in range(i, 5):
        # Обмен элементов по диагонали
        new_matrix[i, j], new_matrix[j, i] = new_matrix[j, i], new_matrix[i, j]
```

Рисунок 3 – Первый блок программного кода

Далее в коде реализован метод Гаусса для нахождения определителя матрицы. Для этого создается копия переданной матрицы matrix, называемая tri_matrix. Это делается для того, чтобы сохранить оригинальную матрицу, так как в процессе работы алгоритма она будет изменяться. В переменную n записывается размерность квадратной матрицы tri_matrix.

После цикл выполняется для каждого элемента на главной диагонали матрицы и проверяется, равен ли текущий элемент на главной диагонали нулю. Если элемент на главной диагонали равен нулю, происходит поиск ненулевого элемента в том же столбце ниже текущей позиции. Если ненулевой элемент не найден, возвращается 0. Иначе происходит обмен строк так, чтобы ненулевой элемент стал на главной диагонали.

В следующем цикле матрица приводится к треугольному виду с помощью элементарных преобразований строк. Вычисляется определитель матрицы, который равен произведению элементов на главной диагонали после приведения к треугольному виду. Функция возвращает вычисленное значение определителя. Далее в коде функция gauss вызывается с новой матрицей new_matrix, и ее результат сохраняется в переменной determinant. Результат представлен ниже (рисунок 4)

```
# Вычисляем определитель матрицы методом Гаусса
2 usages
def gauss(matrix):
    tri_matrix = np.copy(matrix)
    n = len(tri_matrix)

    for i in range(n):
        if tri_matrix[i][i] == 0:
            j = i + 1
            while j < n and tri_matrix[j][i] == 0:
                j += 1
            if j == n:
                return 0
            else:
                tri_matrix[[i, j]] = tri_matrix[[j, i]]

        for j in range(i+1, n):
            r = tri_matrix[j][i] / tri_matrix[i][i]
            tri_matrix[j] -= r * tri_matrix[i]

    result = 1
    for i in range(n):
        result *= tri_matrix[i][i]

    return result

determinant = gauss(new_matrix)
```

Рисунок 4 – Вычисление определителя методом Гаусса

После этого в коде реализовано вычисление алгебраических дополнений. Для этого создается матрица algebraic_complements размером 3x3, заполненная нулями. Для каждого элемента в новой матрице (new_matrix) вычисляется подматрица без строки i и столбца j. Определитель этой подматрицы вычисляется с помощью функции gauss. Вычисляется алгебраическое дополнение для элемента (i, j) как произведение определителя подматрицы на (-1) в степени i + j.

Затем происходит нормализация новой матрицы. В следствие чего, матрица алгебраических дополнений делится на определитель транспонированной матрицы (determinant), полученный на предыдущем этапе. Также мы умножаем матрицу на вектор путем умножения новой матрицы (new_matrix) на вектор-столбец matrix2 с помощью функции np.dot. А также, мы умножаем полученное произведение на транспонированный вектор val_product. Этот результат умножается на исходную матрицу matrix. Результат представлен ниже (рисунок 5).

```python
# Вычисляем алгебраические дополнения
algebraic_complements = np.zeros((5, 5))
for i in range(5):
    for j in range(5):
        # Получение подматрицы
        sub_matrix = np.delete(new_matrix, i, 0)  # Удаляем i-ю строку
        sub_matrix = np.delete(sub_matrix, j, 1)  # Удаляем j-й столбец

        # Вычисление определителя подматрицы
        sub_determinant = gauss(sub_matrix)

        # Вычисление алгебраического дополнения
        algebraic_complements[i, j] = ((-1) ** (i + j)) * sub_determinant

# Делим матрицу алгебраических дополнений на определитель транспонированной матрицы
new_matrix = algebraic_complements / determinant

val_product = np.dot(new_matrix, matrix2)

new_matrix = matrix * val_product.T
```

Рисунок 5 – Алгебраические дополнения и последующие вычисления

После этих математических преобразований создаём пользовательский класс, в котором объявляем конструктор класса. Конструктор принимает на вход три аргумента: new_matrix, val_product и matrix2. Сначала в конструкторе происходит инициализация окон и устанавливается заголовок окна на "Межотраслевый баланс", а также задается размер окна (600x400 пикселей).

Затем мы создаем таблицу, в которой создается виджет QTableWidget, который представляет собой таблицу с ячейками для отображения данных и устанавливаются стили для таблицы с помощью CSS. Это делается, чтобы изменить шрифт, размер текста, цвет фона и другие атрибуты визуального представления таблицы.

После происходит заполнение таблицы. Для этого устанавливается количество строк и столбцов для таблицы. В данном случае 6 строк и 6 столбцов, а значения из матрицы new_matrix и val_product заполняют ячейки таблицы. Это делается вложенными циклами, проходящими по строкам и столбцам таблицы. Для каждой ячейки устанавливается соответствующее значение из матрицы. В конечном итоге, вычисляются суммы по строкам и столбцам. Для строк вычисляется сумма элементов в каждой строке матрицы new_matrix, а для столбцов - сумма элементов в каждом столбце.

После этих операций данные считываются из трех различных Excel-файлов: 'payments_bad.xlsx', 'payments_norm.xlsx' и 'payments_good.xlsx'. Для этого используется библиотека pandas. Для каждого файла вычисляется сумма всех платежей. На основе суммы платежей рассчитываются значения для трех различных прогнозов: пессимистичного, реалистичного и оптимистичного.

Также устанавливаются заголовки для строк и столбцов таблицы. В данном случае строки и столбцы пронумерованы от 1 до 3, а также добавлены заголовки "Сумма", "Конечный продукт" и "Валовый продукт".

Размеры столбцов настраиваются так, чтобы они растягивались на всю доступную ширину окна. Размеры строк настраиваются так, чтобы они автоматически подстраивались под содержимое.

Создается кнопка с надписью "Экспорт в Excel". Настраивается стиль кнопки с использованием CSS и устанавливается обработчик события нажатия на кнопку. Когда кнопка будет нажата, будет вызван метод export_to_excel.

Таблица и кнопка добавляются на главное окно. Для этого сначала создается вертикальный слой, в который добавляются виджет таблицы и кнопка. Затем этот слой устанавливается в качестве центрального виджета главного окна.

Второй метод в классе отвечает за экспорт данных из таблицы PyQt в файл формата Excel. Для этого открывается диалоговое окно для выбора файла Excel. Пользователь выбирает место и имя файла для сохранения. Данные из таблицы PyQt (названия столбцов и содержимое ячеек) конвертируются в формат DataFrame, используя библиотеку pandas.

Названия столбцов и строк в DataFrame устанавливаются на основе заголовков и индексов таблицы PyQt.DataFrame сохраняется в выбранный файл Excel.Индексы и заголовки столбцов также сохраняются в файле. Реализация второго метода класса представлена ниже (рисунок 6).

```
    def export_to_excel(self):
        # Открываем диалог выбора файла для экспорта
        file_name, _ = QFileDialog.getSaveFileName(self, 'Экспорт в Excel', '', 'Excel Files (*.xlsx)')
        if file_name:
            # Конвертируем таблицу PyQt в DataFrame pandas
            data = []
            for row in range(self.table_widget.rowCount()):
                row_data = []
                for column in range(self.table_widget.columnCount()):
                    item = self.table_widget.item(row, column)
                    row_data.append(item.text() if item else '')
                data.append(row_data)
            df = pd.DataFrame(data)

            # Устанавливаем названия столбцов и строк
            headers = [self.table_widget.horizontalHeaderItem(i).text() for i in range(self.table_widget.columnCount())]
            df.columns = headers
            df.index = [self.table_widget.verticalHeaderItem(i).text() for i in range(self.table_widget.rowCount())]

            # Экспортируем DataFrame в Excel с разделителем целой и дробной части - запятая
            df.to_excel(file_name, index=True, header=True)

if __name__ == "__main__":
    app = QApplication(sys.argv)
    window = MatrixTable(new_matrix, val_product, matrix2)
    window.show()
    sys.exit(app.exec_())
```

Рисунок 6 – Реализация второго метода класса и основной функции программы

В ходе написания программного обеспечения были решены многие проблемы. К примеру, корректный импорт данных из Excel. По предварительной оценке, ПО справляется со своими задачами в рамках исследовательской работы и составляет статическую модель межотраслевого баланса на основе полученных данных из файла Excel.

3.2 Тестирование программного обеспечения и вычислительный эксперимент

В начале тестирования ПО проведем функциональное тестирование, которое сосредотачивается на проверке функциональных требований системы и оценивает, насколько программа соответствует ожиданиям пользователя, выполняя проверку функций и возможностей, доступных в приложении. Главная задача такого тестирования – удостовериться, что программное обеспечение ведет себя так, как ожидается со стороны пользователей.

В нашем же случае, важно также проверить корректность полученных данных в ходе математических расчётов.

Произведём тестирование программного обеспечения. Тестирование полученных значений будет производится на основе файла с расчётами в формате Excel.

В первую очередь на вход в систему поступит файл, в котором будет описана матрица коэффициентов прямых затрат и вектор конечной продукции.

Эти данные импортируются в коде, а также будут использоваться для расчётов. Результат представлен ниже (рисунок 7).

	A	B	C	D	E	F	G
1	0,04	0,11	0,29	0,11	0,2		30
2	0,12	0,39	0,15	0,06	0,11		63
3	0,08	0,14	0,24	0,05	0,03		38
4	0,05	0,07	0,11	0,19	0,34		7
5	0,13	0,16	0,12	0,14	0,17		13

Рисунок 7 – Матрица коэффициентов прямых затрат и вектор конечной продукции

При запуске программы перед нами выводится пользовательский интерфейс, в котором мы видим модель межотраслевого баланса. Также мы имеем возможность экспорта представленных значений в виде таблицы в Excel-формате (рисунок 8).

Межотраслевый баланс

	Сельхоз	Хим. Пром.	Металлургия	Энергетика	Машиностроение	Сумма	Конечный продукт	Валовый продукт
Сельхоз	4.52	19.47	30.15	9.43	19.39	82.96	30.0	112.96
Хим. Пром	13.55	69.02	15.59	5.14	10.67	113.98	63.0	176.98
Металлург	9.04	24.78	24.95	4.29	2.91	65.96	38.0	103.96
Энергетика	5.65	12.39	11.44	16.29	32.97	78.73	7.0	85.73
Машиност	14.68	28.32	12.48	12.0	16.48	83.96	13.0	96.96
Сумма	47.44	153.97	94.6	47.15	82.42	425.58	151	576.58
Vj	65.51	23.01	9.36	38.58	14.54			
Валовый п	112.96	176.98	103.96	85.73	96.96			

Экспорт в Excel

Рисунок 8 – Вывод интерфейса

На основе полученной таблицы модели межотраслевого баланса можно провести детальный анализ взаимосвязей между различными производственными и потребляющими отраслями. Модель предоставляет важную информацию о структуре экономики и взаимосвязях между различными секторами.

Сельское хозяйство поставляет значительное количество продукции химической промышленности (19,47) и металлургии (30,15), что свидетельствует о высокой зависимости этих отраслей от сельскохозяйственной продукции. Общая сумма продукции, произведенной сельским хозяйством и потребленной другими отраслями, составляет 82,96, а остальные 30 идут на конечное потребление.

Химическая промышленность имеет значительные поставки в сельское хозяйство (13,55), металлургию (15,59), энергетику (5,14) и машиностроение (10,67).

Металлургия показывает значительную зависимость от химической промышленности, что может указывать на тесные технологические связи. Валовый продукт металлургии составляет 103,96.

Энергетика обеспечивает энергоресурсами все рассмотренные отрасли, при этом наибольший объем продукции идет в машиностроение.

Машиностроение является крупным потребителем продукции других отраслей, особенно энергетики и металлургии. Валовый продукт машиностроения составляет 96,96, но конечное потребление низкое – 13, что может указывать на высокий уровень внутриотраслевого потребления или экспорта.

Общие показатели свидетельствуют о том, что общая сумма продукции, произведенной всеми отраслями для потребления, составляет 425,58, а валовой продукт по всем отраслям – 576,58. Конечное потребление по всем отраслям равно 151. Эти данные свидетельствуют о сильной взаимосвязи и зависимости между отраслями, а также о высоком уровне внутреннего потребления продукции.

Проанализировав результаты, можно сделать вывод, что сельское хозяйство и химическая промышленность играют ключевые роли в межотраслевых связях, с высокими показателями поставок другим отраслям.

Полученные данные могут быть использованы в различных направлениях для улучшения понимания экономических процессов, оптимизации производства и прогнозирования будущих тенденций.

Информация о потоках продукции между отраслями позволяет выявить наиболее важные связи и зависимости. Например, высокая зависимость химической промышленности от сельского хозяйства указывает на необходимость обеспечения стабильных поставок сельскохозяйственной продукции для предотвращения сбоев в химическом производстве. Оптимизация логистических и производственных цепочек на основе этих данных может значительно повысить эффективность работы предприятий.

Результаты анализа межотраслевого баланса могут быть использованы для стратегического планирования и определения приоритетных направлений развития различных отраслей. Например, высокая взаимозависимость металлургии и машиностроения предполагает, что улучшение технологических процессов и инновации в этих секторах могут оказать значительное влияние на всю экономику. Это может включать инвестиции в научные исследования, модернизацию оборудования и повышение квалификации работников.

Модель межотраслевого баланса предоставляет базу для прогнозирования экономических тенденций. Зная текущие взаимосвязи и зависимости между отраслями, можно моделировать влияние различных факторов на экономику в будущем. Например, если планируется увеличение производства в химической промышленности, можно спрогнозировать, как это повлияет на спрос на сельскохозяйственную продукцию и, соответственно, на всю цепочку поставок.

Анализ межотраслевого баланса позволяет оценить, как внешние факторы, такие как изменения в мировой экономике, природные катаклизмы или политические события, могут повлиять на внутреннюю экономику. Если возникает дефицит сельскохозяйственной продукции из-за неблагоприятных погодных условий, можно спрогнозировать, какие отрасли будут затронуты больше всего и какие меры нужно принять для смягчения последствий.

Правительственные органы могут использовать результаты анализа для разработки и корректировки экономической политики. Данные о высоком вкладе металлургии в другие сектора могут стать основанием для стимулирования инвестиций в этот сектор, что приведет к улучшению устойчивости экономики. Также, результаты могут помочь в принятии решений о субсидиях, налоговых льготах и других мерах поддержки.

Понимание межотраслевых связей помогает в управлении рисками, связанными с производством и поставками. Если определённая отрасль сильно зависит от поставок из другого сектора, важно учитывать потенциальные риски, связанные с нарушением этих поставок. Создание запасов, диверсификация поставщиков и разработка альтернативных решений могут помочь минимизировать эти риски.

Компании могут использовать анализ межотраслевого баланса для повышения своей конкурентоспособности. Зная, какие отрасли и предприятия являются ключевыми потребителями их продукции, они могут более точно настраивать свою производственную и маркетинговую стратегию, улучшать качество продукции и сервис, а также устанавливать более тесные партнерские отношения.

Предположим, что правительство планирует значительные инвестиции в развитие энергетической инфраструктуры. Используя данные межотраслевого баланса, можно спрогнозировать, что увеличение инвестиций в энергетику приведет к увеличению спроса на продукцию машиностроения и металлургии, поскольку эти отрасли тесно связаны с энергетикой. Соответственно, можно ожидать роста производства в этих секторах, что в свою очередь может повлиять на всю экономику, создавая новые рабочие места и увеличивая ВВП.

Теперь, когда у нас есть конечный результат, нам необходимо провести вычислительный эксперимент с целью того, чтобы убедиться в правильности полученных результатов. Для этого используем формулы, описанные выше. Сначала, определим, является ли матрица А продуктивной. Для этого вычислим суммы элементов этой матрицы по столбцам. Они будут равны 0,42; 0,87; 0,91; 0,55; 0,85. Так как максимальная сумма 0,91 < 1, то матрица будет продуктивной и решение задачи существует.

Вектор спроса y известен. Задача межотраслевого баланса заключается в определении вектора выпуска x.

Решение уравнения можно представить в виде формулы 8. Для начала находим E – A (рисунок 9).

0,96	-0,11	-0,29	-0,11	-0,2
-0,12	0,61	-0,15	-0,06	-0,11
-0,08	-0,14	0,76	-0,05	-0,03
-0,05	-0,07	-0,11	0,81	-0,34
-0,13	-0,16	-0,12	-0,14	0,83

Рисунок 9 – Решение по формуле 8

Затем вычисляем определитель данной матрицы (формула 18).

$$det(E - A) = 0{,}1962\,. \qquad (18)$$

Теперь находим транспонированную матрицу для матрицы, которая вычислена на рисунке 9. Результат представлен ниже (рисунок 10).

0,96	-0,12	-0,08	-0,05	-0,13
-0,11	0,61	-0,14	-0,07	-0,16
-0,29	-0,15	0,76	-0,11	-0,12
-0,11	-0,06	-0,05	0,81	-0,14
-0,2	-0,11	-0,03	-0,34	0,83

Рисунок 10 – Транспонированная матрица

Находим алгебраические дополнения для элементов транспонированной матрицы. Алгебраические дополнения представлены ниже (рисунки 11-13).

0,61	-0,14	-0,07	-0,16	0,249338
-0,15	0,76	-0,11	-0,12	
-0,06	-0,05	0,81	-0,14	
-0,11	-0,03	-0,34	0,83	

-0,12	-0,08	-0,05	-0,13	0,07931
-0,15	0,76	-0,11	-0,12	
-0,06	-0,05	0,81	-0,14	
-0,11	-0,03	-0,34	0,83	

-0,12	-0,08	-0,05	-0,13	0,047513
0,61	-0,14	-0,07	-0,16	
-0,06	-0,05	0,81	-0,14	
-0,11	-0,03	-0,34	0,83	

-0,12	-0,08	-0,05	-0,13	0,058535
0,61	-0,14	-0,07	-0,16	
-0,15	0,76	-0,11	-0,12	
-0,11	-0,03	-0,34	0,83	

-0,12	-0,08	-0,05	-0,13	0,071084
0,61	-0,14	-0,07	-0,16	
-0,15	0,76	-0,11	-0,12	
-0,06	-0,05	0,81	-0,14	

Рисунок 11 – Алгебраические дополнения для первой группы элементов

-0,11	-0,14	-0,07	-0,16	0,115053	-0,11	0,61	-0,07	-0,16	0,145285
-0,29	0,76	-0,11	-0,12		-0,29	-0,15	-0,11	-0,12	
-0,11	-0,05	0,81	-0,14		-0,11	-0,06	0,81	-0,14	
-0,2	-0,03	-0,34	0,83		-0,2	-0,11	-0,34	0,83	

0,96	-0,08	-0,05	-0,13	0,402342	0,96	-0,12	-0,05	-0,13	0,135933
-0,29	0,76	-0,11	-0,12		-0,29	-0,15	-0,11	-0,12	
-0,11	-0,05	0,81	-0,14		-0,11	-0,06	0,81	-0,14	
-0,2	-0,03	-0,34	0,83		-0,2	-0,11	-0,34	0,83	

0,96	-0,08	-0,05	-0,13	0,098463	0,96	-0,12	-0,05	-0,13	0,310171
-0,11	-0,14	-0,07	-0,16		-0,11	0,61	-0,07	-0,16	
-0,11	-0,05	0,81	-0,14		-0,11	-0,06	0,81	-0,14	
-0,2	-0,03	-0,34	0,83		-0,2	-0,11	-0,34	0,83	

0,96	-0,08	-0,05	-0,13	0,109061	0,96	-0,12	-0,05	-0,13	0,11
-0,11	-0,14	-0,07	-0,16		-0,11	0,61	-0,07	-0,16	
-0,29	0,76	-0,11	-0,12		-0,29	-0,15	-0,11	-0,12	
-0,2	-0,03	-0,34	0,83		-0,2	-0,11	-0,34	0,83	

0,96	-0,08	-0,05	-0,13	0,128212	0,96	-0,12	-0,05	-0,13	0,112358
-0,11	-0,14	-0,07	-0,16		-0,11	0,61	-0,07	-0,16	
-0,29	0,76	-0,11	-0,12		-0,29	-0,15	-0,11	-0,12	
-0,11	-0,05	0,81	-0,14		-0,11	-0,06	0,81	-0,14	

Рисунок 12 – Алгебраические дополнения для второй и третьей группы элементов

-0,11	0,61	-0,14	-0,16	0,070253	-0,11	0,61	-0,14	-0,07	0,109359
-0,29	-0,15	0,76	-0,12		-0,29	-0,15	0,76	-0,11	
-0,11	-0,06	-0,05	-0,14		-0,11	-0,06	-0,05	0,81	
-0,2	-0,11	-0,03	0,83		-0,2	-0,11	-0,03	-0,34	
0,96	-0,12	-0,08	-0,13	0,067083	0,96	-0,12	-0,08	-0,05	0,104826
-0,29	-0,15	0,76	-0,12		-0,29	-0,15	0,76	-0,11	
-0,11	-0,06	-0,05	-0,14		-0,11	-0,06	-0,05	0,81	
-0,2	-0,11	-0,03	0,83		-0,2	-0,11	-0,03	-0,34	
0,96	-0,12	-0,08	-0,13	0,042041	0,96	-0,12	-0,08	-0,05	0,052931
-0,11	0,61	-0,14	-0,16		-0,11	0,61	-0,14	-0,07	
-0,11	-0,06	-0,05	-0,14		-0,11	-0,06	-0,05	0,81	
-0,2	-0,11	-0,03	0,83		-0,2	-0,11	-0,03	-0,34	
0,96	-0,12	-0,08	-0,13	0,291295	0,96	-0,12	-0,08	-0,05	0,15186
-0,11	0,61	-0,14	-0,16		-0,11	0,61	-0,14	-0,07	
-0,29	-0,15	0,76	-0,12		-0,29	-0,15	0,76	-0,11	
-0,2	-0,11	-0,03	0,83		-0,2	-0,11	-0,03	-0,34	
0,96	-0,12	-0,08	-0,13	0,079147	0,96	-0,12	-0,08	-0,05	0,306996
-0,11	0,61	-0,14	-0,16		-0,11	0,61	-0,14	-0,07	
-0,29	-0,15	0,76	-0,12		-0,29	-0,15	0,76	-0,11	
-0,11	-0,06	-0,05	-0,14		-0,11	-0,06	-0,05	0,81	

Рисунок 13 – Алгебраические дополнения для четвертой и пятой группы элементов

На рисунках 11-13 алгебраические дополнения выделены зеленым цветом, а элементы, участвующие в расчётах этих дополнений, выделены желтым цветом.

После нахождения алгебраических дополнений находим искомую матрицу $(E - A)^{-1}$ (формула 19). Численный результат представлен ниже (рисунок 14).

$$(E - A)^{-1} = \frac{1}{det(E - A)} \begin{vmatrix} A_{11} & A_{12} & A_{13} & A_{14} & A_{15} \\ A_{21} & A_{22} & A_{23} & A_{24} & A_{25} \\ A_{31} & A_{32} & A_{33} & A_{34} & A_{35} \\ A_{41} & A_{42} & A_{43} & A_{44} & A_{45} \\ A_{51} & A_{52} & A_{53} & A_{54} & A_{55} \end{vmatrix} . \quad (19)$$

1,27	0,59	0,74	0,36	0,56
0,40	2,05	0,69	0,34	0,53
0,24	0,50	1,58	0,21	0,27
0,30	0,56	0,56	1,48	0,77
0,36	0,65	0,57	0,40	1,56

Рисунок 14 – Нахождение искомой матрицы

Найденная матрица по формуле 19 является матрицей полных затрат. Зная её, мы можем найти вектор выпуска X (формула 20). Численный результат представлен ниже (рисунок 15).

$$X = (E - A)^{-1} * Y \quad . \tag{20}$$

1,27	0,59	0,74	0,36	0,56
0,40	2,05	0,69	0,34	0,53
0,24	0,50	1,58	0,21	0,27
0,30	0,56	0,56	1,48	0,77
0,36	0,65	0,57	0,40	1,56

Рисунок 15 – Нахождение вектора выпуска X

Для нахождения величин межотраслевых потоков необходимо умножить столбцы матрицы A на найденные значения валового выпуска. Также найдем условно-чистую продукцию. В конечном итоге, в ходе математических преобразований полученная таблица модели межотраслевого баланса будет выглядеть следующим образом (рисунок 16).

	Производящие отрасли	Потребляющие отрасли					Сумма	Конечный продукт	Валовый продукт
45–46		Сельское хозяйство	Хим. Пром.	Металлургия	Энергетика	Машиностроение			
47	Сельское хозяйство	4,52	19,47	30,15	9,43	19,39	82,96	30,00	112,96
48	Хим. Пром.	13,55	69,02	15,59	5,14	10,67	113,98	63,00	176,98
49	Метталургия	9,04	24,78	24,95	4,29	2,91	65,96	38,00	103,96
50	Энергетика	5,65	12,39	11,44	16,29	32,97	78,73	7,00	85,73
51	Машиностроение	14,68	28,32	12,48	12,00	16,48	83,96	13,00	96,96
52	Сумма	47,44	153,97	94,60	47,15	82,42	425,58	151,00	576,58
53	Vj	65,51	23,01	9,36	38,58	14,54			
54	Валовый продукт	112,96	176,98	103,96	85,73	96,96			

Рисунок 16 – Таблица межотраслевого баланса

Исходя из рисунка 16 можно сделать вывод, что все вычисления полностью совпали с конечным результатом программы. Соответственно, программа работает стабильно и выводит корректный результат.

Выводы по разделу 3.

В третьем разделе ВКР была подробно описана реализация программного кода.

Также было произведено функциональное тестирование ПО и вычислительный эксперимент, в ходе которого было выяснено, что программа работает корректно и все значения выводит правильно.

Заключение

В ходе выполнения ВКР были поставлены задачи и определены цели ВКР. Также была выбрана область макроэкономических показателей для проведения исследовательской работы на основе модели межотраслевого баланса.

Далее была описана математическая модель межотраслевого баланса. Была выбрана статическая модель в рамках поставленной задачи.

Был произведен выбор языка программирования и платформы для разработки ПО. После чего система была описана по модулям обработки, визуализации и ввода данных, а также была спроектирована блок-схема системы для дальнейшей разработки программного обеспечения.

В заключительном разделе ВКР была подробно описана реализация программного обеспечения и проведено функциональное тестирование ПО, а также был произведён вычислительный эксперимент с целью выявления правильности полученных результатов. В ходе вычислительного эксперимента проблем не было обнаружено. На основе полученных результатов был проведен анализ и прогнозирование.

Цель работы была выполнена: разработано программное обеспечение, которое выводит схему статической модели межотраслевого баланса на основе макроэкономических показателей.

Задачи, определённые для достижения цели работы, были выполнены в полном объёме.

Список используемой литературы

1. Белоусов В.И. Межотраслевые модели экономики / В.И. Белоусов. — М.: Финансы и статистика, 1984. — 240 с.

2. Борисов В.Д. Методы межотраслевого анализа в планировании и управлении народным хозяйством / В.Д. Борисов. — М.: Изд-во МГУ, 1975. — 180 с.

3. Волков В.И., Дергачев П.В. Математические модели в народном хозяйстве / В.И. Волков, П.В. Дергачев. — М.: Изд-во МГУ, 1984. — 280 с.

4. Гончаров В.М., Громаков Б.В. Межотраслевой анализ и планирование: Методические указания и контрольные задания / В.М. Гончаров, Б.В. Громаков. — М.: Финансы и статистика, 1987. — 160 с.

5. Громаков Б.В. Моделирование экономических процессов / Б.В. Громаков. — М.: Изд-во МГУ, 1982. — 200 с.

6. Дергачев П.В., Лосев А.Н. Моделирование в экономике / П.В. Дергачев, А.Н. Лосев. — М.: Финансы и статистика, 2001. — 360 с.

7. Кравченко С.А. Методы исследования и моделирования экономических процессов / С.А. Кравченко. — М.: Изд-во МГУ, 2007. — 250 с.

8. Кудрявцев С.Т. Моделирование экономических процессов / С.Т. Кудрявцев. — М.: Финансы и статистика, 2002. — 310 с.

9. Лосев А.Н. Моделирование экономических процессов / А.Н. Лосев. — М.: Изд-во МГУ, 2001. — 280 с.

10. Макаров В.Л. Межотраслевые модели экономики / В.Л. Макаров. — М.: Изд-во МГУ, 1985. — 220 с.

11. Модель межотраслевого баланса России. — М.: Изд-во МГУ, 2005. — 180 с.

12. Немчинов В.С. Математические модели в экономике / В.С. Немчинов. — М.: Высшая школа, 1989. — 240 с.

13. Петрович В.Г. Методы макроэкономического анализа / В.Г. Петрович. — М.: Финансы и статистика, 1998. — 190 с.

14. Полтерович В.М. Моделирование экономических процессов / В.М. Полтерович. — М.: Изд-во МГУ, 1995. — 320 с.

15. Решетников А.Н. Методы межотраслевого анализа / А.Н. Решетников. — М.: Финансы и статистика, 1983. — 210 с.

16. Спектор А.М. Методы и модели эконометрики / А.М. Спектор. — М.: Финансы и статистика, 1990. — 270 с.

17. Тяпков М.М. Математические методы экономического анализа / М.М. Тяпков. — М.: Финансы и статистика, 2003. — 290 с.

18. Урнов М.Ю. Моделирование экономических процессов / М.Ю. Урнов. — М.: Изд-во МГУ, 2000. — 260 с.

19. Федосеев В. В. Экономико-математические методы и прикладные модели – М.: Юнити, 1999 – 224 - 300 с.

20. Филипчук В.С. Моделирование экономических процессов / В.С. Филипчук. — М.: Финансы и статистика, 1999. — 300 с.

21. Шевелев Ю.П. Математические методы и модели в экономике / Ю.П. Шевелев. — М.: Финансы и статистика, 1988. — 230 с.

22. Arrow, K.J., Chenery, H.B., Minhas, B.S., & Solow, R.M. (1961). Capital-labor substitution and economic efficiency. The Review of Economics and Statistics, 43(3), 225-250.

23. Diewert, W.E. (1971). An application of the Shephard duality theorem: A generalized Leontief production function. Journal of Political Economy, 79(3), 481-507.

24. Hirschman, A.O. (1958). The strategy of economic development. New Haven: Yale University Press.

25. Leontief, W. (1936). Quantitative input and output relations in the economic systems of the United States. The Review of Economics and Statistics, 18(3), 105-125.

26. Miller, R.E., & Blair, P.D. (2009). Input-output analysis: Foundations and extensions. Cambridge University Press.

27. Pyatt, G., & Round, J.I. (1977). Social accounting matrices for development planning. The Review of Income and Wealth, 23(4), 339-364.

28. Sonis, M., Hewings, G.J.D., & Guo, J. (2003). Feedback structures of interregional interindustry dynamics. Economic Systems Research, 15(2), 161-181.

29. Stone, R. (1952). Input-output relations in the construction of integrated economic accounts. The Review of Economic Studies, 20(2), 107-128.

30. Wassily, L. (1949). The structure of the American economy, 1919–1939: An empirical application of equilibrium analysis. New York: Oxford University Press.

Приложение А

Листинг 1 – Реализация программы

```
import sys
from PyQt5.QtWidgets import QApplication, QFileDialog, QPushButton, QHeaderView, QMainWindow, QTableWidget, QTableWidgetItem, QVBoxLayout, QWidget
import pandas as pd
import numpy as np

# Загружаем первую матрицу из Excel файла
matrix = pd.read_excel('matrix.xlsx', header=None, nrows=5, usecols="A:E").values

# Загружаем вторую матрицу из Excel файла
matrix2 = pd.read_excel('matrix.xlsx', header=None, usecols="G:G").values

# Проверяем, что матрица 5x5
if matrix.shape != (5, 5):
    raise ValueError("Матрица должна быть размером 5x5")

new_matrix = np.eye(5)
for i in range(5):
    new_matrix[i, i] = 1 - matrix[i, i]

# Умножаем остальные элементы на -1
for i in range(5):
    for j in range(5):
        if i != j:
```

```
        new_matrix[i, j] = -matrix[i, j]

#Транспонирование матрицы
for i in range(5):
    for j in range(i, 5):
        # Обмен элементов по диагонали
        new_matrix[i, j], new_matrix[j, i] = new_matrix[j, i], new_matrix[i, j]

# Вычисляем определитель матрицы методом Гаусса
def gauss(matrix):
    tri_matrix = np.copy(matrix)
    n = len(tri_matrix)

    for i in range(n):
        if tri_matrix[i][i] == 0:
            j = i + 1
            while j < n and tri_matrix[j][i] == 0:
                j += 1
            if j == n:
                return 0
            else:
                tri_matrix[[i, j]] = tri_matrix[[j, i]]

        for j in range(i+1, n):
            r = tri_matrix[j][i] / tri_matrix[i][i]
            tri_matrix[j] -= r * tri_matrix[i]

    result = 1
```

Продолжение приложения А:

```
    for i in range(n):
        result *= tri_matrix[i][i]

    return result

determinant = gauss(new_matrix)

# Вычисляем алгебраические дополнения
algebraic_complements = np.zeros((5, 5))
for i in range(5):
    for j in range(5):
        # Получение подматрицы
        sub_matrix = np.delete(new_matrix, i, 0)  # Удаляем i-ю строку
        sub_matrix = np.delete(sub_matrix, j, 1)  # Удаляем j-й столбец

        # Вычисление определителя подматрицы
        sub_determinant = gauss(sub_matrix)

        # Вычисление алгебраического дополнения
        algebraic_complements[i, j] = ((-1) ** (i + j)) * sub_determinant

# Делим матрицу алгебраических дополнений на определитель транспонированной матрицы
new_matrix = algebraic_complements / determinant

val_product = np.dot(new_matrix, matrix2)

new_matrix = matrix * val_product.T
```

Продолжение приложения А:

```
class MatrixTable(QMainWindow):
    def __init__(self, new_matrix, val_product, matrix2):
        super().__init__()
        self.setWindowTitle("Межотраслевый баланс")
        self.setGeometry(100, 100, 600, 400)
        self.table_widget = QTableWidget()

        self.table_widget.setStyleSheet("""
                    QTableWidget {
                        font-family: Arial;
                        font-size: 12pt;
                        color: black;
                        background-color: white;
                        selection-background-color: lightblue;
                    }
                    QHeaderView::section {
                        background-color: lightgray;
                        border: 1px solid darkgray;
                        padding: 4px;
                        min-width: 60px;
                    }
                """)
        self.setCentralWidget(self.table_widget)

        self.table_widget.setRowCount(8)  # Уменьшаем количество строк до 4
        self.table_widget.setColumnCount(8)  # Увеличиваем количество столбцов
до 6
```

Продолжение приложения А:

```
# Заполняем таблицу значениями матрицы
for i in range(5):
    for j in range(5):
        self.table_widget.setItem(i, j, QTableWidgetItem(str(new_matrix[i, j].round(2))))
# Добавляем суммы по столбцам
for i in range(5):
    self.table_widget.setItem(5, i, QTableWidgetItem(str(np.sum(new_matrix[:, i]).round(2))))

self.table_widget.setItem(5, 5, QTableWidgetItem(f"{np.sum(new_matrix):.2f}"))

# Добавляем столбец с "Конечным продуктом"
self.table_widget.setItem(0, 7, QTableWidgetItem(str(val_product[0, 0].round(2))))
self.table_widget.setItem(1, 7, QTableWidgetItem(str(val_product[1, 0].round(2))))
self.table_widget.setItem(2, 7, QTableWidgetItem(str(val_product[2, 0].round(2))))
self.table_widget.setItem(3, 7, QTableWidgetItem(str(val_product[3, 0].round(2))))

self.table_widget.setItem(4, 7, QTableWidgetItem(str(val_product[4, 0].round(2))))
self.table_widget.setItem(5, 7, QTableWidgetItem(str(np.sum(val_product).round(2))))
```

```
self.table_widget.setItem(7, 0, QTableWidgetItem(str(val_product[0, 0].round(2))))
self.table_widget.setItem(7, 1, QTableWidgetItem(str(val_product[1, 0].round(2))))
self.table_widget.setItem(7, 2, QTableWidgetItem(str(val_product[2, 0].round(2))))
self.table_widget.setItem(7, 3, QTableWidgetItem(str(val_product[3, 0].round(2))))
self.table_widget.setItem(7, 4, QTableWidgetItem(str(val_product[4, 0].round(2))))


# Добавляем столбец с "Валовым продуктом"
self.table_widget.setItem(0, 6, QTableWidgetItem(str((val_product[0, 0] - np.sum(new_matrix[0, :])).round(2))))
self.table_widget.setItem(1, 6, QTableWidgetItem(str((val_product[1, 0] - np.sum(new_matrix[1, :])).round(2))))
self.table_widget.setItem(2, 6, QTableWidgetItem(str((val_product[2, 0] - np.sum(new_matrix[2, :])).round(2))))
self.table_widget.setItem(3, 6, QTableWidgetItem(str((val_product[3, 0] - np.sum(new_matrix[3, :])).round(2))))
self.table_widget.setItem(4, 6, QTableWidgetItem(str((val_product[4, 0] - np.sum(new_matrix[4, :])).round(2))))
self.table_widget.setItem(5, 6, QTableWidgetItem(str(np.sum(matrix2))))

self.table_widget.setItem(6, 0, QTableWidgetItem(str((val_product[0, 0] - np.sum(new_matrix[:, 0])).round(2))))
self.table_widget.setItem(6, 1, QTableWidgetItem(str((val_product[1, 0] –
```

```
np.sum(new_matrix[:, 1])).round(2))))
        self.table_widget.setItem(6, 2, QTableWidgetItem(str((val_product[2, 0] - np.sum(new_matrix[:, 2])).round(2))))
        self.table_widget.setItem(6, 3, QTableWidgetItem(str((val_product[3, 0] - np.sum(new_matrix[:, 3])).round(2))))
        self.table_widget.setItem(6, 4, QTableWidgetItem(str((val_product[4, 0] - np.sum(new_matrix[:, 4])).round(2))))

        # Добавляем сумму по строкам
        for i in range(5):
            self.table_widget.setItem(i, 5, QTableWidgetItem(str(np.sum(new_matrix[i, :]).round(2))))

        # Добавляем заголовки
        self.table_widget.setHorizontalHeaderLabels(["Сельхоз", "Хим. Пром.", "Металлургия", "Энергетика", "Машиностроение", "Сумма", "Конечный продукт", "Валовый продукт"])
        self.table_widget.setVerticalHeaderLabels(["Сельхоз", "Хим. Пром.", "Металлургия", "Энергетика", "Машиностроение", "Сумма", "Vj", "Валовый продукт"])

        # Настраиваем размер столбцов и строк
        header = self.table_widget.horizontalHeader()
        header.setSectionResizeMode(QHeaderView.Stretch)

        header = self.table_widget.verticalHeader()
        header.setSectionResizeMode(QHeaderView.ResizeToContents)
```

```
    # Создаем кнопку экспорта
    self.export_button = QPushButton('Экспорт в Excel', self)
    self.export_button.setStyleSheet("""
            QPushButton {
                font-family: Arial;
                font-size: 12pt;
                color: white;
                background-color: #007BFF;
                border: none;
                padding: 10px;
                border-radius: 5px;
            }
            QPushButton:hover {
                background-color: #0056b3;
            }
        """)
    self.export_button.clicked.connect(self.export_to_excel)

    # Добавляем кнопку в интерфейс
    self.layout = QVBoxLayout()
    self.layout.addWidget(self.table_widget)
    self.layout.addWidget(self.export_button)
    self.central_widget = QWidget()
    self.central_widget.setLayout(self.layout)
    self.setCentralWidget(self.central_widget)

def export_to_excel(self):
    # Открываем диалог выбора файла для экспорта
```

Продолжение приложения А:

```
        file_name, _ = QFileDialog.getSaveFileName(self, 'Экспорт в Excel', '', 'Excel Files (*.xlsx)')
        if file_name:
            # Конвертируем таблицу PyQt в DataFrame pandas
            data = []
            for row in range(self.table_widget.rowCount()):
                row_data = []
                for column in range(self.table_widget.columnCount()):
                    item = self.table_widget.item(row, column)
                    row_data.append(item.text() if item else '')
                data.append(row_data)
            df = pd.DataFrame(data)

            # Устанавливаем названия столбцов и строк
            headers = [self.table_widget.horizontalHeaderItem(i).text() for i in range(self.table_widget.columnCount())]
            df.columns = headers
            df.index = [self.table_widget.verticalHeaderItem(i).text() for i in range(self.table_widget.rowCount())]
            # Экспортируем DataFrame в Excel с разделителем целой и дробной части – запятая
            df.to_excel(file_name, index=True, header=True)

if __name__ == "__main__":
    app = QApplication(sys.argv)
    window = MatrixTable(new_matrix, val_product, matrix2)
    window.show()
    sys.exit(app.exec_())
```