

УДК 37.026:512:004.43

https://doi.org/10.33619/2414-2948/125/65

РАЗРАБОТКА МЕТОДИЧЕСКОЙ СИСТЕМЫ ОБУЧЕНИЯ АЛГЕБРЕ С ИСПОЛЬЗОВАНИЕМ ВОЗМОЖНОСТЕЙ БИБЛИОТЕК PYTHON

©**Биймурсаева Б. М.**, ORCID: 0009-0006-6085-3488, SPIN-код: 6213-0776,
канд. пед. наук, Кыргызский национальный университет им. Ж. Баласагына,
г. Нарын, Кыргызстан, bijmursaevaburul@gmail.com

©**Бейшеналиева У. У.**, ORCID: 0000-0001-6209-4581, канд. пед. наук,
Кыргызский национальный университет им. Ж. Баласагына,
г. Нарын, Кыргызстан, uka31ster@gmail.com

©**Рысалиева Д. Ш.**, ORCID ID: 0009-0009-8990-624X, Приамурский государственный
университет им. Шолом-Алейхема, г. Нарын, Кыргызстан, drysalieva@bk.ru

DEVELOPMENT OF A METHODOLOGICAL SYSTEM FOR TEACHING ALGEBRA USING PYTHON LIBRARY CAPABILITIES

©**Bijmursaeva B.**, ORCID: 0009-0006-6085-3488, SPIN code: 6213-0776,
Ph.D. ped. Sciences, Kyrgyz National University named after Zh. Balasagyn,
Naryn, Kyrgyzstan, bijmursaevaburul@gmail.com

©**Beishenalieva U.**, ORCID: 0000-0001-6209-4581, Ph.D. ped. sciences,
Kyrgyz National University named after Zh. Balasagyn,
Naryn, Kyrgyzstan, uka31ster@gmail.com

©**Rysalieva D.**, ORCID: 0009-0009-8990-624X, Amur State University
named after Sholem Aleichem, Naryn, Kyrgyzstan, drysalieva@bk.ru

Аннотация. Рассматривается разработка методической системы обучения алгебре с использованием возможностей языка программирования Python и его специализированных библиотек. Показано, как интеграция информационно-коммуникационных технологий в учебный процесс способствует повышению эффективности усвоения алгебраического материала, формированию компетенций и исследовательских навыков студентов. Представлены методические подходы, структурированные в систему, включающую теоретическое обучение, практические задания, проектную деятельность и визуализацию алгебраических объектов с помощью библиотек NumPy, SymPy и Matplotlib. В статье ещё рассматривается вопрос модернизации методики обучения алгебре в условиях цифровой трансформации образования. Авторы обосновывают необходимость перехода от традиционных вычислительных методов к использованию библиотек языка Python (SymPy, NumPy, Matplotlib) как эффективного инструмента формирования математических компетенций. Описывается структура методической системы, включающая символьные вычисления, матричное моделирование и когнитивную визуализацию. Приводятся практические примеры реализации учебных сценариев в среде Jupyter Notebook. В статье представлена инновационная методическая система обучения алгебре, базирующаяся на интеграции языка Python в учебный процесс. Авторы обосновывают использование библиотек SymPy, NumPy и Matplotlib как инструментов развития математических компетенций. Рассматривается переход от рутинных вычислений к исследовательской деятельности в интерактивной среде Jupyter Notebook. Особое внимание уделяется реализации принципа когнитивной визуализации и алгоритмизации обучения. Предложенная система позволяет оптимизировать когнитивную нагрузку обучающихся, смещая акцент на содержательный анализ математических структур. В работе развиваются идеи автора об интерактивности и эстетизации образования. Описанная

методология способствует формированию глубокого понимания абстрактных понятий и развитию алгоритмического мышления студентов в условиях цифровой трансформации высшей школы.

Abstract. The article discusses the development of a methodological system for teaching algebra using the capabilities of the Python programming language and its specialized libraries. It is shown how the integration of information and communication technologies into the educational process contributes to improving the efficiency of mastering algebraic material, the formation of competencies and research skills of students. Methodological approaches are presented, structured into a system that includes theoretical training, practical tasks, project activities, and visualization of algebraic objects using the NumPy, SymPy, and Matplotlib libraries. The article discusses the modernization of algebra teaching methods in the context of the digital transformation of education. The author substantiates the transition from traditional computational methods to the use of Python programming language libraries (SymPy, NumPy, Matplotlib) as an effective tool for developing mathematical competencies. The structure of the methodological system is described, including symbolic computations, matrix modeling, and cognitive visualization. Practical examples of implementing educational scenarios in the Jupyter Notebook environment are provided. The article presents an innovative methodological system for teaching algebra based on the integration of the Python language into the educational process. The authors substantiates the use of SymPy, NumPy, and Matplotlib libraries as tools for developing mathematical competencies. The transition from routine calculations to research activities in the interactive Jupyter Notebook environment is considered. Special attention is paid to the implementation of the cognitive visualization principle and the algorithmization of learning. The proposed system optimizes students' cognitive load, shifting the focus to the substantive analysis of mathematical structures. The work develops the author's ideas on interactivity and the aestheticization of education. The described methodology promotes a deep understanding of abstract concepts and the development of students' algorithmic thinking in the context of the digital transformation of higher education.

Ключевые слова: обучение алгебре, система обучения, информационно-коммуникационные технологии, библиотеки Python, NumPy, SymPy, Matplotlib, визуализация данных.

Keywords: Algebra training, training system, information and communication technologies, Python libraries, NumPy, SymPy, Matplotlib, data visualization.

Современный этап развития школьного и вузовского образования характеризуется стремительной цифровизацией и поиском новых дидактических подходов, способных повысить мотивацию обучающихся. Алгебра, как одна из наиболее абстрактных областей математики, традиционно вызывает трудности из-за высокого уровня формализации и сложности визуализации математических объектов. В этом контексте разработка методической системы обучения алгебре на основе языка программирования Python представляется крайне актуальной задачей. Традиционные методы обучения часто ограничены использованием статических графиков и ручных вычислений, что в условиях ограниченного учебного времени не позволяет глубоко исследовать динамические свойства функций или решать задачи большой размерности. Использование Python, обладающего мощными библиотеками (такими как SymPy для символьных вычислений, NumPy для работы с массивами и Matplotlib или Plotly для визуализации), позволяет сместить акцент с рутинных вычислений на содержательный

анализ математических моделей. Научная новизна и практическая значимость данной работы заключаются в переходе от эпизодического применения компьютера к созданию целостной методической системы. Такая система включает в себя: интерактивное моделирование алгебраических процессов; автоматизацию проверки сложных аналитических преобразований; визуализацию абстрактных понятий (от теории множеств до матричных трансформаций).

Применение Python в обучении алгебре способствует реализации межпредметных связей между математикой и информатикой, формируя у обучающихся навыки алгоритмического мышления и «исследовательского» подхода к решению задач.

Студенты сначала знакомятся с основными определениями матричного анализа, а затем по очереди подходят к доске и выполняют основные матричные операции. Сложности обычно возникают при ручном методе умножения матриц. Освоение умножения матриц происходит примерно к четвертой или пятой попытке. После этого решение становится вычислительно простым. Платформа находится по адресу электронной почты sampus.fa.ru. Стоит отметить, что студенты очень быстро осваивают техническую сторону вычислений [1].

Рассмотрим пример умножения матриц второго порядка, аналогичный примерам, приведённым в работах Ильина В. А., Позняка Э. Г. по курсу линейной алгебры [4].

Функциональный анализ библиотек Python в контексте алгебраического образования. В данном разделе проводится инвентаризация программных средств и их дидактическое сопоставление с разделами алгебры. Исследуется, как технические характеристики библиотек закрывают конкретные пробелы в традиционном обучении:

Символьные вычисления (SymPy): Анализ возможности преодоления «вычислительного барьера». Рассматривается методика использования системы для проверки аналитических выкладок при решении уравнений и упрощении сложных выражений.

Матричные операции и численные методы (NumPy): Обоснование перехода от скалярного мышления к векторному. Исследование эффективности библиотеки при изучении систем линейных уравнений и основ линейных пространств.

Когнитивная визуализация (Matplotlib/Plotly): Определение роли динамических графиков в формировании визуальных образов функций и их свойств.

Проектирование методической системы на базе интерактивных сред. Здесь описывается сама модель обучения. Основной акцент делается не на коде, а на способе организации учебной деятельности. Среда обучения: обоснование использования Jupyter Notebook как платформы для «грамотного обучения», где текст учебника перемежается с исполняемыми ячейками кода. Типология учебных задач: разработка системы задач нового типа (исследовательские задачи, задачи на верификацию гипотез, задачи на оптимизацию). Интерактивный компонент: использование виджетов управления параметрами для организации лабораторных работ по алгебре [2].

Методика реализации межпредметных связей и алгоритмизации. Исследовательский фокус на том, как изучение алгебры через Python формирует универсальные учебные действия (УУД): трансформация математической формулы в программный алгоритм.

Анализ «ошибок реализации» как инструмента глубокого понимания математической логики (почему код выдает ошибку при делении на ноль или работе с вырожденными матрицами).

Апробация и педагогическая оценка эффективности системы. Раздел посвящен анализу результатов внедрения методики (теоретическому или эмпирическому).

Сравнение уровня усвоения абстрактных понятий в контрольной и экспериментальной группах.

Оценка изменения когнитивной нагрузки: снижение временных затрат на рутину и их перераспределение в пользу анализа и интерпретации.

Анализ мотивационного аспекта: влияние использования «профессиональных» инструментов программирования на интерес к изучению предмета.

Методологическая основа исследования базируется на сочетании системно-деятельностного подхода и концепции визуального мышления. В основу предлагаемой методической системы заложен переход от репродуктивного обучения (решение задач по образцу) к исследовательскому моделированию.

1. Этапы проектирования системы. Разработка системы включает три последовательных этапа: концептуально-аналитический (отбор разделов алгебры (линейная алгебра, теория уравнений, функции), где использование Python дает максимальный дидактический эффект); технологический (подбор конкретных библиотек (SymPy, NumPy, Matplotlib) для реализации математических операций и визуализации); дидактический (создание сценариев взаимодействия «студент — компьютер — преподаватель»).

2. Принципы интеграции библиотек Python. Методология использования библиотек строится на следующих принципах:

Принцип прозрачности вычислений (библиотека SymPy): Использование символьных вычислений не как «черного ящика», а как средства верификации. Студент сначала выполняет аналитический вывод вручную, а затем использует код для проверки промежуточных этапов. Это позволяет локализовать ошибки в логических цепочках.

Принцип структурного моделирования (библиотека NumPy): Обучение переходу от записи уравнений в строку к их представлению в виде матриц и векторов. Это формирует навык алгоритмизации и подготовки данных, что критически важно для будущих специалистов в области Data Science.

Принцип когнитивной визуализации (библиотека Matplotlib): Математический объект исследуется через его графическое представление. Методика предполагает использование динамических параметров (интерактивных графиков), что позволяет визуализировать абстрактные свойства, такие как сходимости или непрерывность.

3. Инструментальная среда (Jupyter-методика). Центральным элементом методологии является использование среды Jupyter Notebook. В отличие от традиционных сред программирования, она позволяет реализовать концепцию «Живого учебника».

Теоретический блок: Описание формул и теорем (Markdown).

Исполняемый блок: Код на Python, выполняющий вычисления.

Визуальный блок: Мгновенная генерация графиков или таблиц.

Рефлексивный блок: Задания на изменение кода и анализ полученных отклонений[3].

4. Методы оценки эффективности. Для проверки результативности методической системы предлагается использовать:

Тестирование когнитивных навыков (способность к обобщению и переносу знаний).

Анализ временных затрат на решение комплексных задач (где требуется и расчет, и построение графика).

Анкетирование уровня учебной автономии студентов.

Разработка методической системы обучения алгебре с использованием Python строится на следующих принципах:

1. Компетентностный подход – ориентация на формирование профессиональных и математических компетенций.

2. Активное и проблемное обучение – студенты самостоятельно решают задачи с использованием Python, что повышает аналитические навыки.

3. Интеграция теории и практики – алгоритмы, модели и вычисления выполняются одновременно, укрепляя понимание структуры алгебраических объектов.

4. Использование цифровых технологий – библиотеки Python обеспечивают эффективное выполнение вычислений и визуализацию результатов.

Методическая система включает следующие компоненты:

Теоретический блок – изучение понятий, операций с матрицами, систем линейных уравнений, собственных значений и векторов.

Практический блок – выполнение вычислений с использованием библиотек Python.

Проектный блок – разработка мини-проектов и исследовательских заданий, включающих моделирование алгебраических процессов.

Визуализационный блок – построение графиков функций, отображение матриц и векторов, использование интерактивных элементов.

Практическое применение. Для реализации методологии на конкретном примере возьмем тему «Системы линейных алгебраических уравнений (СЛАУ) и матричные преобразования». Это фундаментальный раздел алгебры, где вычислительная сложность часто затмевает математическую суть. Раскроем эту тему через три этапа нашей методологии: Практическая реализация: Тема «Матрицы и СЛАУ»

Этап 1. Символьный анализ и верификация (SymPy). На этом этапе студент работает с матрицей как с объектом, содержащим переменные или точные рациональные числа. Это позволяет избежать ошибок округления и сосредоточиться на свойствах.

Задача: Найти определитель матрицы и обратную матрицу для системы $Ax = b$.

Методический прием: Студент выполняет расчет вручную (например, методом дополнений), а затем использует код для мгновенной проверки.

Python

```
import sympy as sp
# Создание матрицы с символьными значениями
A = sp.Matrix([[1, 2], [3, 4]])
# Вычисление определителя и обратной матрицы в точном виде
det_A = A.det()
A_inv = A.inv()
print(f"Определитель: {det_A}")
print(f"Обратная матрица: {A_inv}")
```

Результат: Студент видит структуру обратной матрицы без перевода в десятичные дроби, что облегчает понимание алгоритма.

Этап 2. Структурное моделирование и вычисления (NumPy). Когда концепция понята, мы переходим к работе с реальными данными и изучению «матричного подхода» как инструмента.

Задача: Решить систему из 10 уравнений, моделирующую распределение токов в цепи или баланс в экономике. Методический прием: Обучение векторизации. Вместо циклов студент учится записывать решение одной командой $x = A^{-1}b$.

Python

```
import numpy as np
# Коэффициенты системы
A = np.array([[3, 1], [1, 2]])
b = np.array([9, 8])
# Решение системы линейных уравнений
```

```
x = np.linalg.solve(A, b)
print(f"Решение системы: {x}")
```

Исследовательский аспект: Здесь можно обсудить понятие обусловленности матрицы — что произойдет с решением, если один из коэффициентов изменится на 0.0001? (Экспериментальная проверка устойчивости).

Этап 3. Когнитивная визуализация (Matplotlib). Геометрическая интерпретация матрицы как линейного преобразования пространства. Это «высший пилотаж» понимания алгебры.

Задача: Визуализировать, как матрица A трансформирует единичный квадрат (векторы базиса).

Методический прием: Создание интерактивного графика, где изменение элементов матрицы «на лету» деформирует плоскость.

```
Python
import matplotlib.pyplot as plt
# Исходные векторы (базис)
v1 = np.array([1, 0])
v2 = np.array([0, 1])
# Линейное преобразование (матрица)
T = np.array([[2, 1], [0.5, 1]])
# Трансформированные векторы
v1_t = T @ v1
v2_t = T @ v2
# Визуализация (код отрисовки векторов)
plt.quiver([0, 0], [0, 0], [v1_t[0], v2_t[0]], [v1_t[1], v2_t[1]],
           angles='xy', scale_units='xy', scale=1, color=['r', 'b'])
plt.xlim(-1, 4); plt.ylim(-1, 4); plt.grid()
plt.show()
```

Результат: Студент воочию видит, что такое «растяжение», «сдвиг» и почему при определителе, равном 0, пространство «схлопывается» в линию (вырожденная матрица)[5].

Математика является неотъемлемой частью подготовки современных профессионалов. Математика и информационные технологии, в том числе использование языка программирования Python, являются взаимоисключающими, и правильная организация учебного процесса повышает эффективность изучения и понимания каждой дисциплины до такой степени, что это необходимо учитывать. В условиях оперативной информатизации общества использование компьютерных технологий при преподавании математики открывает новые возможности и дает возможность представить учебный материал в интерактивной и наглядной форме.

Применение информационных технологий в сфере образования приводит к повышению эффективности и оптимизации учебного процесса, применение инновационных форм в образовании позволяет повысить эффективность обучения, интеллектуальный уровень обучаемых, привить навыки самообразования и самоорганизации. Применение информационных технологий позволяет разнообразить учебный процесс и улучшить качество образования на всех его этапах [2].

Современное образование предъявляет новые требования к качеству подготовки специалистов, ориентированных на применение информационных технологий в профессиональной деятельности. В контексте математического образования алгебра занимает ключевое место, формируя у студентов аналитическое мышление, логические навыки и способность к системному анализу.

Использование языка программирования Python и его научных библиотек позволяет: интегрировать активные и интерактивные методы обучения; создавать визуализации и симуляции алгебраических процессов; проводить самостоятельные исследовательские работы студентов; повысить мотивацию и интерес к изучению дисциплины.

Программа Anaconda представляет собой комплексный набор инструментов для работы с библиотеками Python, каждая библиотека содержит группу функций и инструментов, предназначенных для решения определённых задач. Например, работы с большими объёмами данных, астрономических вычислений, обработки изображений, создания и обучения нейронных сетей, выполнения инженерных расчётов, статистического анализа и многого другого. В Anaconda представлено около 1500 библиотек различной тематики — от научных исследований до инженерных приложений [3].

В последнее время мы постоянно используем программное обеспечение Anaconda на практике при выполнении лабораторных работ.

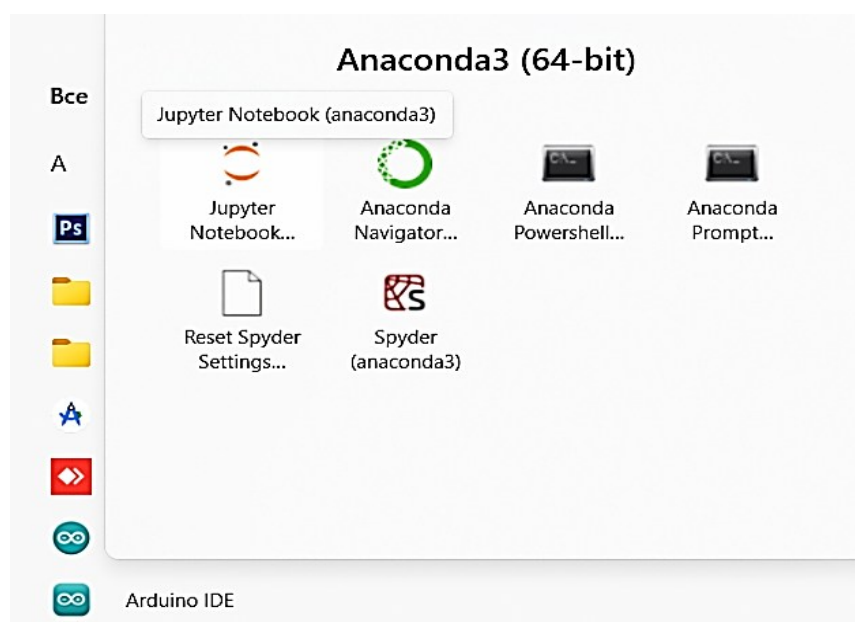


Рисунок 1. Anaconda 3

Jupyter Notebook — это веб-оболочка для IPython (ранее назывался IPython Notebook), представляющая собой веб-приложение с открытым исходным кодом, которое позволяет создавать и совместно использовать документы, включающие в себя функционирующий код, формулы, визуализацию данных и форматированный текст.

Библиотека NumPy предоставляет средства для работы с матрицами, массивами и векторами, выполнения линейных операций, вычисления детерминантов и обратных матриц. Применение NumPy позволяет студентам сосредоточиться на анализе результатов, освобождая от рутинных вычислений.

SymPy обеспечивает возможности символической алгебры, включая: аналитическое решение систем уравнений; упрощение выражений; нахождение собственных значений и собственных векторов.

Использование SymPy способствует глубокому пониманию структуры алгебраических объектов и развитию исследовательских навыков. Библиотека Matplotlib используется для визуализации функций, матриц и векторов.

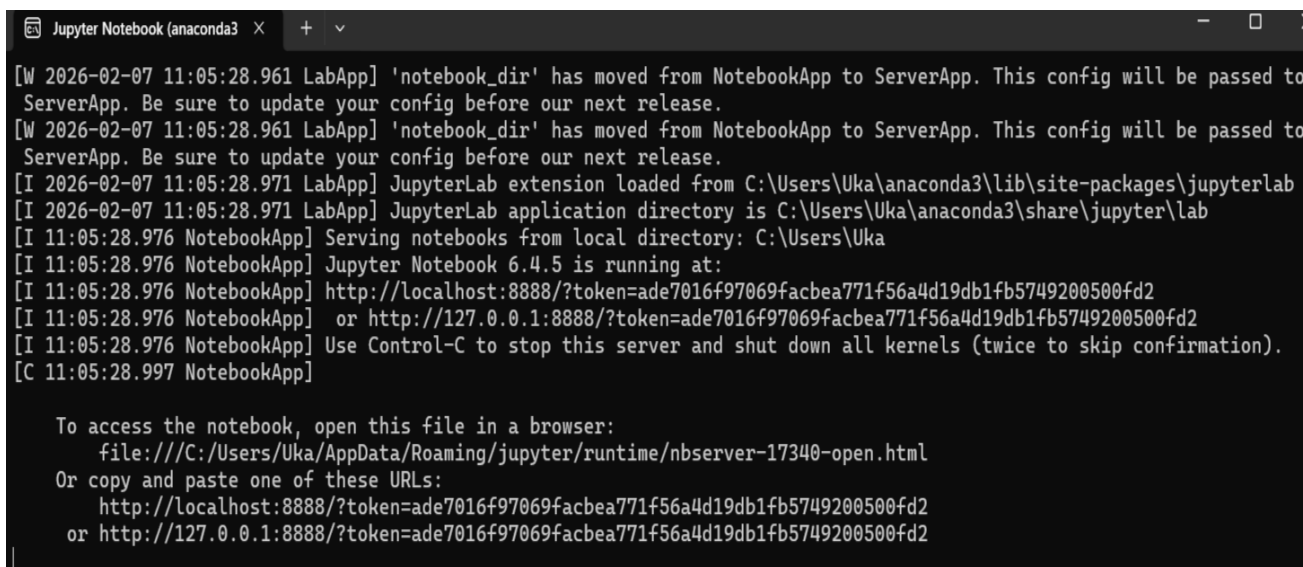
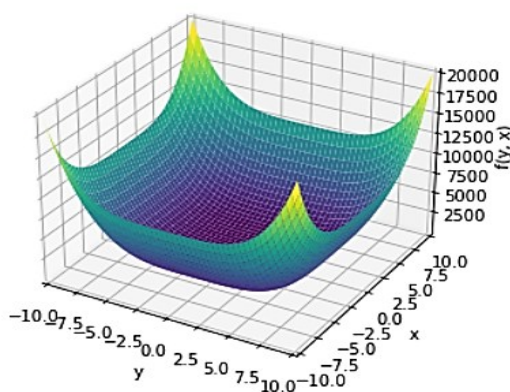


Рисунок 2. Jupyter Notebook

Графическая демонстрация алгебраических процессов повышает наглядность и облегчает восприятие абстрактного материала. Для использования `plot3d()`, требуется задать функцию $z = f(x, y)$, значения границ и шаг изменения переменных x и y . Функция `plot3d()` даёт наглядное представление о поверхности, что может быть полезно при анализе и исследовании функций, особенно если они имеют сложные графики. Работая с данной функцией можно легко исследовать области максимумов и минимумов, а также выявлять другие особенности функции. В результате выполнения программного кода создаются трехмерные графика [3].

```
In [18]: from sympy import*
from sympy.abc import*
from sympy.plotting import*
f = lambda x, y: x**4 + y**4
plot3d(f(x,y))
```



```
Out[18]: <sympy.plotting.plot.Plot at 0x294a40e75e0>
```

Рисунок 3. Трёхмерные графика

В этих условиях студенты могут овладеть алгебраическим и аналитическим мышлением, а также навыками компьютерных вычислений и моделирования. Например, выполнение алгебраических операций с помощью Python, таких как умножение матриц, решение систем

линейных уравнений, вычисление собственных значений и векторов, может помочь студентам укрепить связь между теорией и практикой, обеспечивая исследовательское и проблемное обучение. Кроме того, библиотеки языка программирования Python – NumPy, Matplotlib и SymPy – эффективное и решать математические уравнения алгебраические различных вопросов, а также анализ полученных результатов позволяет интерпретациялоо и визуальный вид. В результате, процесс обучения активных, интерактивных и компетентности, направленные на то, стимулированию студентов глубоких знаний в освоении. Таким образом, интеграция языка программирования Python и компьютерных технологий в алгебраические предметы рассматривается как эффективный метод обучения алгебраическому содержанию современных педагогических технологий и способствует развитию исследовательских и аналитических навыков студентов. Пусть заданы матрицы:

$$A = \begin{pmatrix} 1 & 3 \\ 2 & 4 \end{pmatrix}, \quad B = \begin{pmatrix} 0 & 5 \\ 1 & 2 \end{pmatrix}$$

Требуется вычислить произведение матриц:

$$C = A \cdot B$$
$$A = \begin{pmatrix} 1 & 3 \\ 2 & 4 \end{pmatrix}, \quad B = \begin{pmatrix} 0 & 5 \\ 1 & 2 \end{pmatrix}$$

Мы хотим вычислить $C = A \cdot B$.

Шаг 1. Элемент C_{11} :

$$C_{11} = 1 * 0 + 3 * 1 = 0 + 3 = 3$$

Шаг 2. Элемент C_{12} :

$$C_{12} = 1 * 5 + 3 * 2 = 5 + 6 = 11$$

Шаг 3. Элемент C_{21} :

$$C_{21} = 2 * 0 + 4 * 1 = 0 + 4 = 4$$

Шаг 4. Элемент C_{22} :

$$C_{22} = 2 * 5 + 4 * 2 = 10 + 8 = 18$$

Итоговая матрица:

$$C = \begin{pmatrix} 3 & 11 \\ 4 & 18 \end{pmatrix}$$

Использование библиотеки NumPy позволяет автоматизировать вычисления, снизить вероятность вычислительных ошибок и сосредоточить внимание обучающихся на анализе структуры и свойств алгебраических объектов, что особенно важно при формировании исследовательских и аналитических навыков студентов.

В ходе исследования была разработана и обоснована методическая система обучения алгебре с использованием возможностей языка программирования Python и его специализированных библиотек. Показано, что интеграция информационно-коммуникационных технологий в образовательный процесс позволяет повысить качество усвоения алгебраического материала за счёт сочетания теоретической подготовки, практико-ориентированных заданий, проектной деятельности и визуализации математических объектов.

Использование библиотек NumPy, SymPy и Matplotlib способствует формированию у обучающихся устойчивых математических и цифровых компетенций, развитию аналитического и алгоритмического мышления, а также исследовательских навыков.

```
import numpy as np

A = np.array([[1, 3],
              [2, 4]])
B = np.array([[0, 5],
              [1, 2]])

C = np.dot(A, B)

print("Умножения матриц A * B")
print(C)
```

Рисунок 4. Умножения матриц на языке Python

Применение программной среды Python обеспечивает наглядность изучаемых алгебраических процессов, автоматизацию вычислений и возможность проведения самостоятельных экспериментов, что положительно влияет на мотивацию студентов к изучению дисциплины.

Список литературы:

1. Борисова Л. Р. Некоторые аспекты обучения математике студентов первого курса гуманитарных направлений в экономическом вузе // Проблемы преподавания высшей математики и информатики в условиях новой образовательной парадигмы: Материалы Международной научно-практической конференции. Минск, 2024. С. 6-8.
2. Биймурсаева Б. М., Алсеитова К. А. Интерактивный метод обучения математики в вузе // Научное обозрение. Педагогические науки. 2016. №2. С. 25-29.
3. Иванов Б. Н. Дискретная математика и программирование. М.: КноРус, 2021. 320 с.
4. Медведева Н. В. Инновационный подход к управлению системой образования // Материалы Ивановских чтений. 2012. №2. С. 96-100.
5. Роберт И. В. Теория и методика информатизации образования (психолого-педагогический и технологический аспекты). М.: БИНОМ. 2014. 398 с.
6. Тихонов Д. В., Ермоленко М. О. Современные технологии прикладного программирования и обработки данных. Ярославль, 2024. 232 с.
7. Ильин В. А., Позняк Э. Г. Линейная алгебра. М.: Наука, 1984. 294 с.

References:

1. Borisova, L. R. (2024). Nekotorye aspekty obucheniya matematike studentov pervogo kursa gumanitarnykh napravlenii v ekonomicheskom vuze. In *Problemy prepodavaniya vysshei matematiki i informatiki v usloviyakh novoi obrazovatel'noi paradigmy: Materialy Mezhdunarodnoi nauchno-prakticheskoi konferentsii, Minsk*, 6-8. (in Russian).
2. Biimursaeva, B. M., & Alseitova, K. A. (2016). Interaktivnyi metod obucheniya matematiki v vuze. *Nauchnoe obozrenie. Pedagogicheskie nauki*, (2), 25-29. (in Russian).
3. Ivanov, B. N. (2021). *Diskretnaya matematika i programmirovanie*. Moscow. (in Russian).
4. Medvedeva, N. V. (2012). Innovatsionnyi podkhod k upravleniyu sistemoi obrazovaniya. *Materialy Ivanovskikh chtenii*, (2), 96-100. (in Russian).
5. Robert, I. V. (2014). *Teoriya i metodika informatizatsii obrazovaniya (psikhologo-pedagogicheskii i tekhnologicheskii aspekty)*. Moscow. (in Russian).

6. Tikhonov, D. V., & Ermolenko, M. O. (2024). *Sovremennye tekhnologii prikladnogo programmirovaniya i obrabotki dannykh. Yaroslavl'*. (in Russian).

7. Il'in, V. A., & Poznyak, E. G. (1984). *Lineinaya algebra. Moscow*. (in Russian).

Поступила в редакцию
10.02.2026 г.

Принята к публикации
17.02.2026 г.

Ссылка для цитирования:

Биймурсаева Б. М., Бейшеналиева У. У., Рысалиева Д. Ш. Разработка методической системы обучения алгебре с использованием возможностей библиотек Python // Бюллетень науки и практики. 2026. Т. 12. №4. С. 490-500. <https://doi.org/10.33619/2414-2948/125/65>

Cite as (APA):

Biymursaeva, B., Beishenalieva, U., & Rysalieva, D. (2026). Development of a Methodological System for Teaching Algebra using Python Library Capabilities. *Bulletin of Science and Practice*, 12(4), 490-500. (in Russian). <https://doi.org/10.33619/2414-2948/125/65>