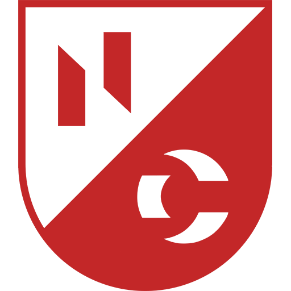
**Некоммерческое образовательное учреждение  
Учебно-научно-производственный комплекс**

**«Международный Университет Кыргызстана»**

**Среднее профессиональное образование**

**«Nomad» Колледж**



**Жоомартбеков Туголбай Бактыбекович**

**КУРСОВАЯ РАБОТА**

*На тему:* «Работа с библиотекой SymPy для решения математических уравнений и символьных вычислений»

Руководитель курсовой работы:

преп.

\_\_\_\_\_\_\_\_\_\_\_ Уралбек уулу С.

“\_\_\_”\_\_\_\_\_\_\_\_\_\_\_\_2024 г.

**Бишкек, 2024 г.**

**Содержание:**

[Введение 3](#_Toc155101543)

[Цель курсовой работы 4](#_Toc155101544)

[ГЛАВА 1. Что такое SymPy? 4](#_Toc155101545)

[1.1 Основные характеристики SymPy 5](#_Toc155101546)

[ГЛАВА 2. Аналоги SymPy 6](#_Toc155101547)

[2.2 Сравнение библиотек 7](#_Toc155101548)

[2.2 Преимущества использования библиотеки SymPy 9](#_Toc155101549)

[2.3 Где применяется SymPy? 11](#_Toc155101550)

[Заключение 13](#_Toc155101551)

[Список используемой литературы 14](#_Toc155101552)

[Приложение 15](#_Toc155101553)

# **Введение**

В современном мире науки и технологий, где сложные вычисления стали неотъемлемой частью исследований и инженерной деятельности, инструменты для символьных математических вычислений занимают особое место. Среди многочисленных библиотек, предназначенных для работы с символами и уравнениями, выделяется SymPy – мощное средство для символьной математики на языке программирования Python.

В своей сути, SymPy представляет собой инструмент, который позволяет разработчикам и математикам проводить вычисления, оперируя символами и выражениями в их аналитической форме, вместо использования чисел.

Это отличает SymPy от традиционных численных методов и придает ему преимущество при работе с точными математическими операциями. Эта библиотека, написанная на языке программирования Python, является открытым программным обеспечением и предоставляет разносторонний инструментарий для работы с символьной математикой.

В данной работе мы рассмотрим SymPy, исследуя его возможности, характеристики и сферы применения. Рассматривая основные аспекты этой библиотеки, мы постараемся понять, как SymPy обеспечивает гибкость и точность символьных вычислений, а также какие преимущества и вызовы существуют при использовании этого инструмента в различных областях науки и инженерии.

Курсовая работа охватит ключевые аспекты использования SymPy, начиная с основных операций и заканчивая более сложными математическими проблемами. В результате исследования мы получим понимание функциональных возможностей SymPy и раскроем потенциал данной библиотеки в решении разнообразных задач, стимулируя интерес к символьным вычислениям и их практическому применению.

# **Цель курсовой работы**

Цель нашей курсовой работы – рассмотреть основные возможности SymPy, выявить области ее применения и проанализировать преимущества символьных вычислений по сравнению с численными методами. Мы также рассмотрим примеры использования SymPy в решении конкретных математических задач, демонстрируя, как библиотека может быть полезным инструментом для улучшения точности и понимания результатов вычислений.

**ГЛАВА 1. Что такое SymPy?**

**SymPy** — это Python библиотека для символьных математических вычислений, которая предоставляет мощные инструменты для работы с символьными выражениями, уравнениями и системами уравнений. Она позволяет разработчикам и исследователям создавать и анализировать модели физических систем с высокой степенью точности и гибкости.

SymPy основана на концепции символьных вычислений, что означает, что она оперирует символами и символьными переменными вместо чисел. Это позволяет выполнять точные математические операции без потери точности из-за аппроксимации. Этот подход особенно полезен при моделировании сложных физических явлений, где аналитические решения могут быть сложными или даже невозможными.

SymPy включает в себя функции базовой символьной [арифметики](https://ru.wikipedia.org/wiki/%D0%90%D1%80%D0%B8%D1%84%D0%BC%D0%B5%D1%82%D0%B8%D0%BA%D0%B0), [математический анализ](https://ru.wikipedia.org/wiki/%D0%9C%D0%B0%D1%82%D0%B5%D0%BC%D0%B0%D1%82%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B8%D0%B9_%D0%B0%D0%BD%D0%B0%D0%BB%D0%B8%D0%B7), [алгебру](https://ru.wikipedia.org/wiki/%D0%90%D0%BB%D0%B3%D0%B5%D0%B1%D1%80%D0%B0) и [дискретную математику,](https://ru.wikipedia.org/wiki/%D0%94%D0%B8%D1%81%D0%BA%D1%80%D0%B5%D1%82%D0%BD%D0%B0%D1%8F_%D0%BC%D0%B0%D1%82%D0%B5%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0) элементы [квантовой физики](https://ru.wikipedia.org/wiki/%D0%9A%D0%B2%D0%B0%D0%BD%D1%82%D0%BE%D0%B2%D0%B0%D1%8F_%D0%BC%D0%B5%D1%85%D0%B0%D0%BD%D0%B8%D0%BA%D0%B0).

### 1.1 Основные характеристики SymPy

SymPy предоставляет обширный набор функций, среди которых основные характеристики включают:

**Символьные переменные:** Основой SymPy являются символьные переменные. Это позволяет пользователю работать с математическими выражениями в их символьной форме, не прибегая к численной аппроксимации. Введение символов и символьных выражений в коде Python делает возможным аналитическое решение уравнений и проведение других символьных операций. [Смотреть рис. 1](#_Приложение)

**Уравнения и системы уравнений:** SymPy предоставляет функции для решения уравнений и систем уравнений. Это включает в себя как линейные, так и нелинейные уравнения. Результаты представляются в виде символьных выражений. [Смотреть рис. 2](#_Приложение)

Это означает, что единственное решение системы уравнений x + y = 5 и x - y = 3 равно {x: 4, y: 1}. Таким образом, x равно 4, а y равно 1, удовлетворяя обоим уравнениям системы.

**Математический анализ:** SymPy обеспечивает функции для дифференцирования и интегрирования символьных выражений. Это позволяет проводить аналитический анализ функций и получать точные результаты.

**Алгебраические операции:** Библиотека поддерживает операции алгебры, такие как упрощение выражений, факторизация, разложение на множители и другие. [Смотреть рис. 3](#_Приложение).

**Дискретная математика:** SymPy включает в себя инструменты для работы с комбинаторикой, теорией чисел, логикой и другими областями дискретной математики.

В библиотеке SymPy, функция binomial(n, k) представляет собой биномиальный коэффициент. Биномиальный коэффициент C(n,k)C(n,k), также известный как "n по k" или "n choose k", определяется следующим образом: C(n,k)=n!/k!(n−k)!, где n! обозначает факториал числа n.

Таким образом, функция binomial(n, k) возвращает значение биномиального коэффициента для заданных значений n и k.

Это вычисляет биномиальный коэффициент для C(5,2), что равно 10. [Смотреть рис. 4](#_Приложение)

**Элементы квантовой физики:** SymPy предоставляет базовую поддержку для символьного моделирования элементов квантовой физики, таких как операторы и волновые функции. [Смотреть рис. 5](#_Приложение)

# **ГЛАВА 2. Аналоги SymPy**

Существует несколько библиотек и пакетов в языке программирования Python, которые предоставляют символьные вычисления и математические возможности, аналогичные SymPy. Вот несколько из них:

**SymEngine:** это библиотека символьных вычислений на C++, которую можно использовать в Python через оболочку. SymEngine ставит своей целью обеспечение высокой производительности символьных вычислений.

**Maxima:** Maxima — это система компьютерной алгебры (CAS), аналогичная SymPy. Она предоставляет символьные вычисления и широкий набор математических возможностей.

**Mathics:** Mathics — это символьная математическая система и язык программирования, который пытается предоставить функциональность, аналогичную Mathematica.

**SageMath:** SageMath — это математический пакет, объединяющий множество библиотек и инструментов, включая SymPy. Он предоставляет обширный функционал для численных и символьных вычислений.

**Mathematica (WolframScript):** предоставляет средства для выполнения математических вычислений с использованием языка Mathematica. Это мощный инструмент с символьными и численными возможностями.

**CAS (Computer Algebra System) в Python:** Некоторые другие пакеты и библиотеки также предоставляют символьные вычисления в Python. Например, mpmath, sympygamma и другие.

Выбор конкретной библиотеки зависит от конкретных требований и предпочтений, таких как производительность, доступность функций, интеграция с другими библиотеками и т.д.

### 2.1 Сравнение библиотек

Выбор между библиотеками для символьных вычислений зависит от конкретных требований проекта, и каждая из них имеет свои сильные стороны. Рассмотрим преимущества использования библиотеки SymPy в сравнении с некоторыми из упомянутых выше альтернатив:

**SymPy vs. SymEngine:**

**SymPy:** SymPy имеет высокий уровень доступности для пользователей благодаря интеграции с Jupyter Notebook и удобному интерфейсу в Python. Его использование также распространено в образовательных целях.

**SymEngine:** SymEngine ориентирован на высокую производительность, но его API написан на C++. Если вам нужна максимальная скорость символьных вычислений и вы готовы пожертвовать удобством использования, SymEngine может быть интересным выбором.

**SymPy vs. Maxima:**

**SymPy:** Maxima — это система компьютерной алгебры, в то время как SymPy предоставляет символьные вычисления как часть библиотеки на Python. SymPy может быть более удобен для интеграции в проекты на Python.

**Maxima:** Maxima может предоставлять более широкий набор функций в определенных областях и может быть предпочтителен в контексте, где требуется максимальная функциональность компьютерной алгебры.

**SymPy vs. SageMath:**

**SymPy:** SymPy является частью SageMath, предоставляя символьные вычисления в рамках обширной математической системы. Он также может быть использован самостоятельно в проектах, не требующих всех возможностей SageMath.

**SageMath:** SageMath объединяет множество библиотек и инструментов в обширную математическую систему, предоставляя полный стек математических инструментов.

**SymPy vs. Mathematica (WolframScript):**

**SymPy:** SymPy является свободной и открытой библиотекой, доступной бесплатно. Он хорошо интегрирован с экосистемой Python и обладает широкой поддержкой.

**Mathematica (WolframScript):** Mathematica является коммерческим продуктом и может предоставлять более широкие возможности в определенных областях, но зачастую сопряжено с коммерческой лицензией.

**SymPy vs. CAS в Python:**

**SymPy:** SymPy является стандартным выбором для символьных вычислений в экосистеме Python. Он широко используется, обладает активным сообществом и хорошей документацией.

**CAS в Python:** Некоторые другие пакеты, такие как mpmath, могут предоставлять символьные вычисления, но часто они ограничены в функциональности по сравнению с SymPy.

Выбор зависит от конкретных требований проекта, уровня поддержки, доступности функций и других факторов. SymPy обычно выбирается благодаря своей свободной и открытой природе, активному сообществу, удобству использования и интеграции с языком Python.

### 2.2 Преимущества использования библиотеки SymPy

**Точность:** в отличие от численных методов, SymPy работает с символами, обеспечивая высокую степень точности в вычислениях.

**Гибкость:** Библиотека предоставляет широкий спектр математических функций, что делает ее универсальным инструментом для различных областей науки и инженерии.

**Образовательная ценность:** SymPy часто используется в образовательных целях для демонстрации концепций символьных вычислений и аналитической математики.

**Интеграция с Python:** благодаря интеграции с языком программирования Python, SymPy легко внедряется в проекты, использующие этот язык.

**Открытый исходный код:** SymPy является проектом с открытым исходным кодом, что позволяет пользователям изучать, модифицировать и вносить свои вклады в развитие библиотеки.

**Совместимость с LaTeX:** SymPy обладает встроенной поддержкой LaTeX, что делает возможным создание красиво оформленных математических выражений для использования в научных статьях, презентациях и документации.

**Интерактивное использование:** благодаря интеграции с средой Jupyter Notebook, SymPy может использоваться интерактивно, что делает его удобным инструментом для обучения, экспериментов и исследовательской работы.

**Интеграция с другими библиотеками:** SymPy хорошо интегрируется с другими научными библиотеками Python, такими как NumPy и SciPy. Это обеспечивает возможность комбинировать символьные вычисления с численными методами для более широкого спектра задач.

**Графическое представление:** SymPy позволяет визуализировать математические выражения с использованием графических библиотек, таких как Matplotlib, что облегчает визуализацию результатов.

**Многозадачность:** благодаря своей модульной структуре, SymPy позволяет решать разнообразные задачи в области математики и физики. Он может использоваться для символьного интегрирования, дифференцирования, решения уравнений, алгебраических манипуляций и даже символьного решения дифференциальных уравнений.

**Поддержка расширений:** SymPy предоставляет возможность создания пользовательских функций и расширений, что позволяет пользователям внедрять свои символьные операции и алгоритмы.

**Активное сообщество:** SymPy разрабатывается и поддерживается активным сообществом разработчиков и математиков. Это обеспечивает постоянное обновление и улучшение библиотеки, а также поддержку пользователей.

### 2.3 Где применяется SymPy?

SymPy обладает широким спектром применения в различных областях науки, инженерии и образования. Ниже представлены некоторые из сфер применения библиотеки SymPy:

**Научные исследования:** SymPy активно используется в научных исследованиях, особенно в области математики, физики и инженерии. Он предоставляет удобные средства для символьных вычислений, которые могут быть важными при решении аналитических задач и проведении символьных вычислений.

**Образование:** SymPy широко используется в образовательных целях. Он помогает студентам изучать и понимать математические концепции, проводить символьные вычисления и решать уравнения. Интеграция SymPy с Jupyter Notebook делает его отличным выбором для обучения.

**Инженерия и технические вычисления:** SymPy применяется в инженерных расчетах, моделировании и анализе систем. Он может использоваться для символьного решения уравнений, работы с алгебраическими выражениями и проведения различных символьных операций.

**Математическое моделирование:** SymPy предоставляет инструменты для создания и анализа математических моделей различных систем, включая физические, экономические и биологические системы.

**Физика:** В области физики SymPy используется для символьного моделирования физических законов, решения уравнений движения, работы с волновыми функциями в квантовой механике и других задач.

**Статистика и анализ данных:** SymPy может применяться для работы с символьными выражениями в статистике и анализе данных. Например, проведение символьных вычислений в рамках статистических задач.

**Тестирование и верификация программ:** SymPy может быть использован для генерации и проверки символьных выражений в ходе тестирования программного обеспечения, особенно в случае, когда требуется точность символьных вычислений.

# **Заключение**

В ходе выполнения данной курсовой работы были изучены основные аспекты библиотеки SymPy — мощного инструмента для символьных математических вычислений на языке программирования Python. SymPy представляет собой проект, основанный на концепции символьных вычислений, позволяя оперировать символами и символьными переменными для точных математических операций без потери точности.

Цель использования SymPy заключается в предоставлении разработчикам, исследователям и студентам инструмента для работы с символьными выражениями, уравнениями и системами уравнений. Библиотека призвана облегчить аналитические решения математических задач, где численные методы могут быть недостаточно точны или неудобны в использовании.

На фоне рассмотрения альтернативных библиотек, таких как SymEngine, Maxima, Mathics и других, SymPy выделяется своей простотой использования, интеграцией с Python, активным сообществом и обширным функционалом для символьных вычислений.

Преимущества SymPy включают в себя не только возможность решения уравнений, систем уравнений и проведения символьного анализа, но также широкий спектр функций, таких как математический анализ, алгебра, дискретная математика и элементы квантовой физики.

Сферы применения SymPy охватывают научные исследования, образование, инженерные расчеты, математическое моделирование, физику, статистику, а также тестирование и верификацию программ. Благодаря своей универсальности, библиотека SymPy становится неотъемлемым инструментом в разнообразных областях, где требуются символьные вычисления для точного и гибкого решения математических задач.

# **Список используемой литературы**

1. <https://pythonru.com/biblioteki/sympy-v-python>
2. <https://ru.wikipedia.org/wiki/SymPy>
3. <https://habr.com/ru/companies/otus/articles/761344/>
4. <https://pythonist.ru/biblioteka-sympy-simvolnye-vychisleniya-v-python/>
5. <https://www.sympy.org/ru/features.html>
6. <https://habr.com/ru/articles/423731/>
7. <https://hashdork.com/ru/sympy-library-guide/>
8. <https://pypi.org/project/symengine/>
9. <https://www.altlinux.org/Images/0/0b/MaximaBook.pdf>
10. <https://informatics.msk.ru/mod/book/view.php?id=4293&chapterid=460>
11. <https://docs.python.org/3/library/math.html>
12. <https://dev-gang.ru/article/sagemath-zanimaemsja-matematikoi-na-python-nw0cm2djzy/>
13. <https://www.wolfram.com/mathematica/new-in-8/dynamic-library-loading/index.html.ru?footer=lang>

Ссылка на Online Jupiter Notebooks:

https://cocalc.com/features/jupyter-notebook

Ссылка на GitHub:

https://github.com/TYGOLKA/kyrsovaya/tree/main

# **Приложение**

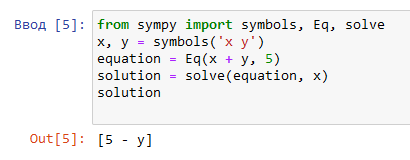


Рис. 1

from sympy import symbols, Eq, solve

x, y = symbols('x y')

equation = Eq(x + y, 5)

solution = solve(equation, x)

solution

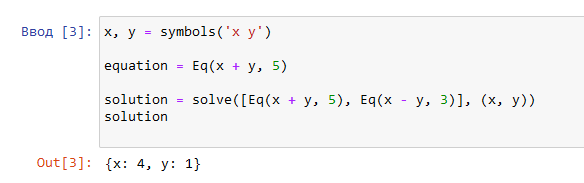


Рис. 2

from sympy import symbols, Eq, solve

x, y = symbols('x y')

equation = Eq(x + y, 5)

solution = solve([Eq(x + y, 5), Eq(x - y, 3)], (x, y))

solution

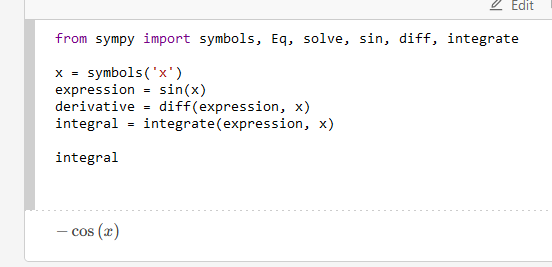


Рис. 3

from sympy import symbols, Eq, solve, sin, diff, integrate

x = symbols('x')

expression = sin(x)

derivative = diff(expression, x)

integral = integrate(expression, x)

integral

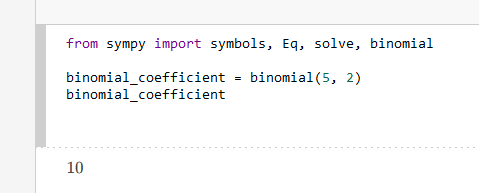


Рис. 4

from sympy import symbols, Eq, solve, binomial

binomial\_coefficient = binomial(5, 2)

binomial\_coefficient

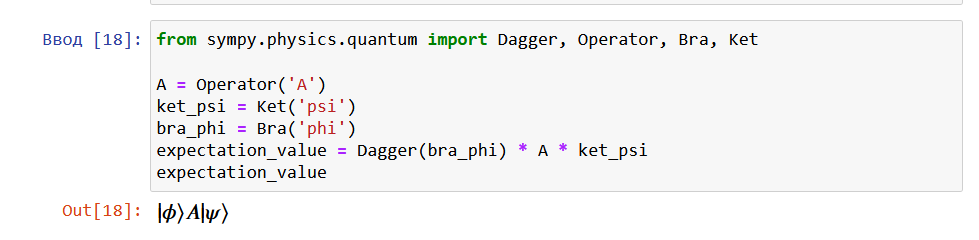


Рис. 5

from sympy.physics.quantum import Dagger, Operator, Bra, Ket

A = Operator('A')

ket\_psi = Ket('psi')

bra\_phi = Bra('phi')

expectation\_value = Dagger(bra\_phi) \* A \* ket\_psi

expectation\_value