ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ
ФГАОУ ВО НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

Факультет компьютерных наук
Образовательная программа «Прикладная математика и информатика»

Отчет о программном проекте

на тему Интегрирование в sympy с использованием тригонометрических подстановок

Выполнил:
студент группы БПМИ 196 _____ Д. О. Рубин
Подпись И.О. Фамилия

29.05.2021
Дата

Принял:
руководитель проекта Жукова Галина Николаевна
Имя, Отчество, Фамилия
доцент, доцент
Должность, ученое звание
ДПИ ФКН
Место работы (Компания или подразделение НИУ ВШЭ)

Дата проверки 29.05 2021 8 _____
Оценка (по 10-тибалльной шкале) Подпись

Москва 2021

Реферат

Программный проект посвящён теме реализации алгоритмов замены переменных для интегрирования с использованием тригонометрических подстановок в библиотеке символьных вычислений языка Python3 Sympy. В рамках данного проекта изучены и описаны достоинства и недостатки существующих решений. Аналоги: функция trigintegrate, интегрирующая тригонометрические функции, используя шаблоны; метод transform для замены переменных в определенных и неопределенных интегралах. На основе анализа имеющихся алгоритмов реализован метод trig_transform класса Integral, выполняющий необходимые преобразования для замены переменных. Для метода написана документация и docstring на английском языке для интеграции в библиотеку SymPy.

Реализованный метод trig_transform используя встроенные в SymPy функции solve и trigsimp вычисляет выражение функции, дифференциал и пределы интегрирования для проведения замены переменных.

Метод trig_transform реализован на языке программирования Python3.8. Тестирование проходило на ОС Linux с использованием примеров из курса математического анализа.

Ключевые слова: Python3, SymPy, trig_transform, transform, Тригонометрическая подстановка, Замена переменных в интеграле

Оглавление

Основные термины и определения

1. SymPy – библиотека языка Python с открытым исходным кодом для символьных вычислений.
2. Замена переменных в интеграле – такая замена, при которой x меняется на $f(t)$, а dx меняется на $f'(t)dt$
3. Универсальная тригонометрическая подстановка – замена в интеграле вида$t = tg(x/2)$. При такой подстановке $dx = \frac{2dt}{1+t^2}$, $sin(x) = \frac{2t}{1+t^2}$, $cos(x) = \frac{1-t^2}{1+t^2}$,$x = arctg(t)$
4. Тригонометрическая подстановка $x = asin(t)$для интегралов вида $\int \sqrt{(a^2 - x^2)}\, dx$
5. Тригонометрическая подстановка $x = atg(t)$для интегралов вида $\int \sqrt{(x^2 + a^2)}\, dx$
6. Тригонометрическая подстановка $x = \frac{a}{sin(t)}$для интегралов вида $\int \sqrt{(x^2 - a^2)}\, dx$

Введение

В наше время компьютеризированные вычисления играют очень большую роль во всех областях жизни, в особенности в науке. Они используются в авиации, космонавтике, архитектуре. И если в 20-м веке до развития вычислительных систем необходимые расчёты приходилось проводить руками, то сейчас для этого используются компьютеры.

Преимущества очевидны: компьютеры не допускают ошибок, им не требуется отдых, они могут перебрать тысячи способов решения задачи за короткий временной промежуток. Именно поэтому с развитием вычислительных мощностей и популяризацией компьютеров человечество перешло на компьютеризированные вычисления.

Однако, компьютеры всё ещё ограничены в том плане, что они могут проводить только алгоритмические вычисления. То есть им необходим алгоритм или программа, следуя которым они будут проводить расчёты. Поэтому, людям все ещё необходимо принимать участие для написания этих алгоритмов для компьютеров.

Так, например, библиотека символьных вычислений SymPy языка Python позволяет вычислять определённые и неопределённые интегралы, но с ограничениями. SymPy позволяет проводить замены переменных в интеграле, однако только такие, при которых между старой и новой переменной можно установить взаимно однозначное соответствие. Поэтому библиотека не позволяет считать интегралы с помощью тригонометрических подстановок, широко используемых в математическом анализе.

Цель данного проекта – реализовать метод (функцию класса) trig_transform для проведения тригонометрических подстановок в определённом и неопределённом интеграле. Написать для него документацию и примеры использования по образцу, принятому в SymPy.

Обзор и сравнительный анализ источников и аналогов

- **Метод transform [1.b] –** заменяет переменную в интеграле, пересчитывая пределы интегрирования (для определённых интегралов), используя встроенные функции библиотеки sympy
  - Позволяет производить только такие замены переменных, при которых между новым и старым значением можно установить взаимноодноозначное соответствие.
  - Не позволяет провести замену $x^2 - 1, u$в интеграле $\int (x^2 - 1)\, dx$
- **Встроенный в библиотеку SymPy метод trigintegrate**
  - Позволяет вычислять интегралы от тригонометрических функций.
  - trigintegrate использует шаблоны для интегрирования, поэтому может вычислять только интегралы от конкретных функций и их линейных комбинаций, таких как, например $\int sin\,(x)cos(x)dx$
  - Поскольку используются шаблоны, trigintegrate не способен вычислить, например, интеграл $\int \left(\frac{1}{5-3cos(x)}\right) dx$

  Но всё вышеописанное трудно назвать аналогами trig_integrate, из-за отличий в заявленном функционале. На самом деле, аналогов trig_integrate не существует, потому что SymPy единственная крупная библиотека для символьных вычислений в Python. Поэтому аналогами могут быть только функции из SymPy, перечисленные выше.

Описание функциональных требований

Требования к функциональным характеристикам:

- Реализован метод **trig_transform** класса **Integral [1.a]**
  - **trig_transform** принимает два аргумента – x (старая переменная), u (новая переменная – выражение)
- Дополнительно реализован (вспомогательный) метод **_calc_limits** для пересчёта пределов интегрирования в определённых интегралах

Требования к организации входных данных:

- Корректные переменные для замены (переменные – символы)
  - Поддерживаются функции вида:
    - $x = asin(u)$
    - $x = acos(u)$
    - $x = atg(u)$
    - $x = \frac{a}{sin(u)}$

Требования к организации выходных данных:

- Интеграл с корректно проведённой заменой переменных
- Новые пределы интегрирования (если есть)
- Тот же самый интеграл, если замену провести нельзя
- Исключение **ValueError** в случае некорректного ввода или входных данных

Требования к временным характеристикам:

- Не предъявляются (в связи с особенностями SymPy)

Описание нефункциональных требований

- Требования к документации
  - doc_string – на английском
    - Описание исключений и ошибок, которые могут возникать в trig_transform
    - Примеры правильного и неправильного использования trig_transform
  - Краткое описание метода trig_transform и его возможностей
- Требования к примерам
  - Примеры применения замены переменной к типовым задачам из курса математического анализа
- Требования к тестам
  - Корректные тесты на каждую из тригонометрических подстановок
  - Тесты, некорректно использующие метод trig_transform

При этом, при некорректных тестах метод trig_transform должен сообщать об ошибке, а при обычных тестах выдавать правильный ответ.

- Дополнительные требования
  - Параметр debug, при котором пользователю выводится выражение, дифференциал новой переменной, и новые пределы интегрирования

Реализация

Поскольку конечной целью проекта предполагается интегрирование метода **trig_transform** в библиотеку символьных вычислений **SymPy [2]** языка **Python [3],** метод реализован на языке Python3.8.

Краткое описание алгоритма работы метода **trig_transform**:

- В качестве входных переменных метод принимает два выражения (встроенный в **SymPy** класс **Expression [1]**): x, u, где x – заменяемое выражение, а u – заменяющее выражение, при этом выполняется равенство
  - $x = f(u)$
- Проводятся необходимые проверки на корректность входных данных, возможные исключения:
  - Заменяемая переменная не совпадает с переменной интегрирования
    - Например, при замене $t,\ 2\sin(u)$ в интеграле $\int f(x)dx$
  - Число свободных переменных для замены больше одной
  - «Старая» переменная не является символом
  - «Новая» переменная не является символом
  - «Новая» переменная содержит несколько свободных переменных
- Получаем интеграл с заменёнными переменными, используя встроенную в **SymPy** функцию **subs [1]**, заменяющую переменные в выражении
- Упрощаем полученный интеграл встроенной функцией **trigsimp [1.c]**
- Для неопределённых интегралов возвращаем полученный интеграл в качестве результата
- Для определённых интегралов пересчитываем пределы интегрирования функцией **__calc_limits**
- Функция **__calc_limits**
  - Используя встроенную функцию **solve** выражает u через g(x), используя соотношение $x = f(u)$
  - Проверяет корректность пределов для подстановок
    - $asin(t)\ acos(t)\ atg(t)$
  - Пересчитывает пределы используя g(x), полученную функцией **solve**
- Возвращаем определенный интеграл с проведённой подстановкой

Примеры проведения замены в интегралах

Приведем примеры проведения тригонометрической замены в интегралах [4]

1. Замена вида $x = 2sin(t)$

$$I = \int \left(\frac{\sqrt{(4 - x^2)}}{x} \right) dx = \begin{bmatrix} x = 2 * sin(t) \\ dx = 2 * cos(t) * dt \\ 3 \end{bmatrix} = 2 * \int \left(\frac{cos^2(t)}{sin(t)} \right) dt;$$

2. Замена вида $x = tg(t)$

$$I = \int \left(\frac{dx}{(1 + x^2)^3} \right) = \begin{bmatrix} x = tg(t) \\ dx = \frac{dt}{cos^2(t)} \end{bmatrix} = \int \left(\frac{\frac{dt}{cos^2(t)}}{\left(\sqrt{\frac{1}{cos^2(t)}} \right)^3} \right) = \int cos\,(t) * dt$$

3. Замена вида $x = \frac{2}{sin(t)}$

$$I = \int \left(\frac{x}{\sqrt{x^2 - 4}} \right) dx = \int \left(\frac{\frac{-2}{sin^3(t)} * cos(t)}{\sqrt{\frac{4}{sin^2(t)} - 4}} \right) * dt = \int \left(\frac{-2}{sin^2(t)} \right)$$

Результат работы программы для одного из примеров выше:

[7]:
```
from sympy import Integral
from sympy.abc import x
from sympy import sqrt, sin
from sympy import Symbol

t = Symbol('t', real=True)

I = Integral(x / sqrt(x ** 2 - 4), x)
I
```

[7]:
$$\int \frac{x}{\sqrt{x^2 - 4}}\, dx$$

[8]:
```
I.trig_transform(x, 2 / sin(t))
```

[8]:
$$\int \left(-\frac{2}{\sin^2(t)} \right) dt$$

Результаты

Реализовано:

1. Основной код метода trig_transform
2. Обработка исключений:
   - Исходная переменная не совпадает с переменной интегрирования
   - Число свободных переменных для замены больше одной
   - «Старая» переменная не является символом
   - «Новая» переменная не является символом
   - «Новая» переменная содержит несколько свободных переменных
3. Быстрая замена в случае, когда «Старая» переменная совпадает с «Новой» переменной
4. Написана документация для метода trig_transform
5. Написаны примеры использования для метода trig_transform
6. Оформлены примеры замены переменных из курса Математического Анализа
7. Написан docstring для интеграции кода в sympy
8. Добавлен параметр debug для пошаговой замены переменных

Исходный код проекта: **GitHub**

Заключение

В рамках данного программного проекта, посвящённого тригонометрическим подстановкам в интегралах, был реализован метод trig_transform класса Integral, в библиотеке символьных вычислений SymPy Python3. Алгоритм для метода был придуман на основе анализа аналогов. Написана документация для метода, doc_string для интеграции в исходный код библиотеки SymPy, примеры корректного и некорректного использования, описаны исключения, возникающие при работе метода.

На данном этапе реализован полностью работающий и готовый к использованию метод. В перспективе можно было бы добавить отсечку по времени работы, оптимизировать код, написав аналоги встроенных в SymPy методов (например **solve**), более подходящие для использования в trig_transform.

Список источников

1. **Документация библиотеки SymPy** – [Электронный ресурс]
    a. **Документация класса Integral** – [Электронный ресурс]
    b. **Документация метода transform** – [Электронный ресурс]
    c. **Документация метода trigsimp** – [Электронный ресурс]
2. **Исходный код библиотеки SymPy –** [Электронный ресурс]
3. **Документация Python3.8** – [Электронный ресурс]
4. **Основы математического анализа**. В 2-х ч. **Ильин В. А., Позняк Э. Г.** М.: Физматлит. Ч.2 - 2005, 7-е изд., 648с.