

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 сумру с использованием тригонометрических подстановок
(промежуточный, этап 1)

Выполнил:

студент группы БПМИ 196


Подпись

Д. О. Рубин
И.О. Фамилия

08.02.2021

Дата

Принял:

руководитель проекта

Жукова Г.Н.

Имя, Отчество, Фамилия

к.ф.-м.н., доцент

Должность, ученое звание

ДПН ФКН

Место работы (Компания или подразделение НИУ ВШЭ)

Дата проверки 22.02 2021

7 (хорошо)

Оценка
(по 10-тибалльной шкале)


Подпись

Москва 2021

Оглавление

Основные термины и определения.....	2
Введение.....	3
Обзор и сравнительный анализ источников и аналогов.....	4
Описание функциональных требований.....	5
Описание нефункциональных требований.....	6
Список источников.....	7
Приложение.....	8

Основные термины и определения

1. SymPy – библиотека языка Python с открытым исходным кодом для символьных вычислений.
2. Универсальная тригонометрическая подстановка – замена в интеграле вида $t = \tan(x/2)$. При такой подстановке $dx = \frac{2dt}{1+t^2}$, $\sin(x) = \frac{2t}{1+t^2}$, $\cos(x) = \frac{1-t^2}{1+t^2}$, $x = \arctan(t)$
3. Тригонометрическая подстановка $x = a \sin(t)$ для интегралов вида $\int \sqrt{a^2 - x^2} dx$
4. Тригонометрическая подстановка $x = a \tan(t)$ для интегралов вида $\int \sqrt{x^2 + a^2} dx$
5. 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-ом 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**
 - Позволяет производить только такие замены переменных, при которых между новым и старым значением можно установить взаимнооднозначное соответствие.
 - Не позволяет провести замену $x^2 - 1$, и 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, таких как $\int \sin(x)\cos(x)dx$
 - Поскольку используются шаблоны, trigintegrate не способен вычислить, например, интеграл $\int \left(\frac{1}{5-3\cos(x)}\right) dx$

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`
 - `trig_transform` принимает два аргумента – `x` (старая переменная), `u` (новая переменная)
- Дополнительно реализован метод `_calc_limits` для пересчёта пределов интегрирования в определенных интегралах

Требования к организации входных данных:

- Корректные переменные для замены

Требования к организации выходных данных:

- Интеграл с корректно заменёнными переменными
- Правильные пределы интегрирования
- Исключение `ValueError` в случае ошибки с описанием ошибки при замене или пересчёте пределов интегрирования

Требования к временным характеристикам:

- Не предъявляются

Описание нефункциональных требований

- Требования к документации
 - Краткое описание метода `trig_integrate` и его возможностей
 - Описание исключений и ошибок, которые могут возникать в `trig_transform`
 - Примеры правильного и неправильного использования `trig_transform`
- Требования к примерам
 - Примеры с понятно написанным ходом и алгоритмом решения из курса математического анализа
- Требования к тестам
 - Тесты на каждую из тригонометрических подстановок
 - Тесты, некорректно использующие метод `trig_integrate`

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

- Дополнительные требования
 - Параметр `debug`, позволяющий увидеть пошаговую замену переменных в интеграле

Список источников

- *Документация библиотеки SymPy* – [Электронный ресурс] <https://docs.sympy.org>
- *Документация Python3.8* – [Электронный ресурс] <https://docs.python.org/3.8/>
- **Основы математического анализа.** В 2-х ч. Ильин В.А., Позняк Э.Г. М.: Физматлит. Ч.2 - 2005, 7-е изд., 648с.

Приложение

Календарный план

Запланированные задачи	Срок выполнения
Изучение области исследования	1 ноября — 20 декабря
Изучение SymPy	11 – 20 января
Анализ существующих решений	20 — 31 января
Написание КТ-1	1 февраля — 8 февраля
Написание метода <code>trig_integrate</code> : <ul style="list-style-type: none">• Функция пересчёта пределов для определённого интеграла• Возможные исключения метода <code>trig_integrate</code>• Метод <code>trig_integrate</code>• Параметр <code>debug</code>	24 февраля — 31 марта
Подбор примеров из курса математического анализа с алгоритмом решения	1 апреля – 10 апреля
Написание документации по образцу, принятому в SymPy	10 апреля — 30 апреля