



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ «Информатика и системы управления»

КАФЕДРА "Программное обеспечение ЭВМ и информационные технологии"

ОТЧЁТ
К ЛАБОРАТОРНОЙ РАБОТЕ №1
НА ТЕМУ:
“Разработка ПО Интегральный калькулятор”

Студент ИУ7-68Б(В)
(Группа)

(Подпись, дата)

Д.П. Косаревский
(И.О.Фамилия)

Преподаватель

(Подпись, дата)

В.И. Солодовников
(И.О.Фамилия)

2021 г.

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

УТВЕРЖДАЮ
Заведующий кафедрой ИУ7
(Индекс)
И.В.Рудаков
(И.О.Фамилия)
« ____ » _____ 2021 г.

З А Д А Н И Е
на выполнение лабораторной работы

по дисциплине «Основы программной инженерии»

Студент группы ИУ7-68Б(В)

Косаревский Дмитрий Петрович

(Фамилия, имя, отчество)

Тема лабораторной работы «Разработка ПО»

Задание:

1. Написать программу для приближенного вычисления определенного интеграла методами трапеций с заданным шагом, трапеций с заданной точностью, а также точки пересечения выбранной пользователем функции с осью абсцисс на заданном интервале методом дихотомии. Выбранную для проверки функцию передавать как отдельный параметр подпрограмм вычисления значений функции, интеграла, корня.

Для проверки использовать следующие функции:

1) $\sin(x)$;

2) $\cos^2(x) \cdot \ln^2(x + 5)$.

2. Разработать спецификацию требований (техническое задание) на программный продукт.

3. Нарисовать блок-схемы алгоритмов, реализованных в программном продукте

Дата выдачи задания « ____ » _____ 2021 г.

Преподаватель

(Подпись, дата)

В.И. Солодовников

(И.О.Фамилия)

Студент

(Подпись, дата)

Д.П. Косаревский

(И.О.Фамилия)

Спецификация требований к программному обеспечению (Software Requirements Specification) на основе стандарта ISO/IEC/IEEE 29148:2011

История изменений

Дата	Версия	Описание	Автор
23.03.2021	0.1	Документ создан	Косаревский Д.П.

Утверждения

Кто утвердил	ФИО	Дата	Подпись
Гейткипер	****		
Ответственный за проект	****		
Разработчик	Д. Косаревский		
Разработчик	***		

- **Введение**

- Цели

В данном документе подробно описываются все внешние проявления и сценарии поведения разрабатываемого в рамках проекта “Интегральный калькулятор” (далее «система») приложения (или его части, подсистемы). Наряду с этим приводится перечень нефункциональных требований, проектных ограничений и других аспектов, необходимых для полного и всестороннего описания всех требований участников к проектному решению.

- Соглашения о терминах

Сокращение русское	Сокращение английское	Пояснение
<i>Система</i>		<i>Совокупность логически связанных между собой объектов, организованных некоторым образом в единое целое.</i>
<i>Поле (свойство, атрибут)</i>		<i>Характеристика объекта, низшая неделимая единица системы.</i>
<i>Экранная форма (экран)</i>		<i>Совокупность элементов интерфейса пользователя, выводимых на экран одновременно с возможностью просмотра содержимого без дополнительной навигации.</i>

Представление		Совокупность настроек внешнего вида экранной формы. В отношении перечня пользовательское представление – это набор и порядок отображаемых полей.
Пользователь		Пользователь, имеющий доступ на просмотр определенной информации в системе

- Предполагаемая аудитория и последовательность восприятия

Аудиторией будут являться преподаватели и студенты МГТУ им. Баумана, а также любой иной пользователь, который получит доступ к системе.

- Масштаб проекта

Проект является учебным в рамках лабораторных работ по курсу “Основы программной инженерии” в МГТУ им. Баумана.

- Ссылки на источники

Система размещена в сети по адресу:

https://share.streamlit.io/dkosarevsky/sef_lab/main/integral_trapezoidal.py

- **Общее описание**

- Видение продукта

Основные задачи системы:

- *Вычисление определённого интеграла методами трапеций с заданным шагом*
- *Вычисление определённого интеграла методами трапеций с заданной точностью*
- *Вычисление точки пересечения функции с осью абсцисс на заданном интервале методом дихотомии*

- Функциональность продукта

В зависимости от типа вычисления и выбранной задачи пользователь может производить параметрическую настройку вычислений, например:

- *Нижний и верхний предел интегрирования*
- *Шаг разбиений*
- *Необходимую точность вычислений*
- *Начало и конец интервала*

- Классы и характеристики пользователей

Чёткой классификации пользователей не предусмотрено, пользователем может являться любой человек, никаких ограничений не накладывается.

- Среда функционирования продукта (операционная среда)

Система работает из web-интерфейса, работоспособность без ограничений по ОС будет доступна при наличии соединения с интернет. Также возможен локальный запуск, при этом система будет запущена в веб-странице браузера.

- Рамки, ограничения, правила и стандарты

При локальном запуске присутствует ряд ограничений, а именно необходима установка ЯП Python версии не ниже чем 3.8.8, а также библиотек, перечень которых определён в файле requirements.txt в репозитории проекта.

- Документация для пользователей

Проект не является настолько масштабным, чтобы писать полноценную документацию, однако предусмотрены как текстовые описания необходимых действий внутри приложения для рядовых пользователей, так и комментарии и docstrings в коде для продвинутых пользователей.

- Допущения и зависимости

Для корректной работы системы есть ряд обязательных зависимостей в виде python-библиотек:

streamlit==0.79.0

numpy==1.20.1

plotly==4.14.3

- **Функциональность системы**

- Функциональный блок 1

- Описание и приоритет

Приближенное вычисление определенного интеграла методом трапеций с заданным шагом. Высокий приоритет.

- Функциональные требования

Должна быть реализована возможность параметризации расчета с помощью ввода верхнего и нижнего предела интегрирования, а также шага разбиения.

- Функциональный блок 2

- Описание и приоритет

Приближенное вычисление определенного интеграла методом трапеций с заданной точностью. Высокий приоритет.

- Функциональные требования

Должна быть реализована возможность параметризации расчета с помощью ввода верхнего и нижнего предела интегрирования, а также точности вычисления.

- Функциональный блок 3

- Описание и приоритет

Вычисление точки пересечения функции с осью абсцисс на заданном интервале методом дихотомии. Высокий приоритет.

- Функциональные требования

Должна быть реализована возможность параметризации расчета с помощью ввода начала и конца интервала, а также шага значения .

- **Требования к внешним интерфейсам**

- Интерфейсы пользователя (UX)

Пользовательские интерфейсы включают в себя переключатели между необходимыми видами вычислений в виде радиокнопок:

Выберите необходимое вычисление

- ☐ 1. Приближенное вычисление определенного интеграла методом трапеций с заданным шагом
- ☐ 2. Приближенное вычисление определенного интеграла методом трапеций с заданной точностью
- ☒ 3. Вычисление точки пересечения функции с осью абсцисс на заданном интервале методом дихотомии

При выборе какого-либо из видов вычислений пользовательские интерфейсы включают параметризацию вычислителя, для 1-го вычисления:

Приближенное вычисление определенного интеграла методом трапеций с заданным шагом

Введите нижний предел: Введите верхний предел: Введите шаг:

0.00	-	+	1.57	-	+	0.01000000	-	+
------	---	---	------	---	---	------------	---	---

для 2-го вычисления:

Приближенное вычисление определенного интеграла методом трапеций с заданной точностью

Введите нижний предел: Введите верхний предел: Введите точность:

0.00	-	+	1.57	-	+	0.10000000	-	+
------	---	---	------	---	---	------------	---	---

для 3-го вычисления:

Вычисление точки пересечения функции с осью абсцисс на заданном интервале методом дихотомии

Начало интервала (a): Конец интервала (b): Эпсилон (e):

-5.00	-	+	12.50	-	+	0.00001000	-	+
-------	---	---	-------	---	---	------------	---	---

○ Программные интерфейсы

Интерфейс системы должен обеспечивать простоту, скорость и точность ввода информации, наглядность представления данных, отсутствие утомляемости при длительной работе пользователя. Данные условия определяют следующие требования:

- ★ Наглядность, понятность, динамичность интерфейса
- ★ Отсутствие излишних отвлекающих элементов
- ★ Оптимальные размеры элементов управления
- ★ Оптимальный подбор цветовой гаммы элементов интерфейса

○ Интерфейсы связи и коммуникации

Интерфейс коммуникации с пользователем при выборе вычисления точки пересечения функции с осью абсцисс должен обеспечивать отображение интерактивных графиков на которых будут отображены как сами функции, так и точка пересечения функции с осью абсцисс.

- **Нефункциональные требования**

- Требования к производительности

Система не предъявляет специальных требований к производительности.

- Требования к сохранности (данных)

Система не предъявляет требований к сохранности данных. Отсутствуют взаимодействия с БД, никакие данные не сохраняются.

- Требования к качеству программного обеспечения

- Система должна быть надёжной

- ◇ Устойчива к отказам

- ◇ Способность к восстановлению работоспособности при отказах

- Практичность, удобство использования

- ◇ Понятность

- ◇ Простота и удобство использования (требование относится к интерфейсу)

- Эффективность

- ◇ Временные характеристики

- ◇ Использование ресурсов

- Сопровождаемость

- ◇ Анализируемость

- ◇ Изменяемость

- ◇ Контролируемость

- Переносимость, мобильность

- ◇ Адаптируемость

- ◇ Удобство установки

- Требования к безопасности системы

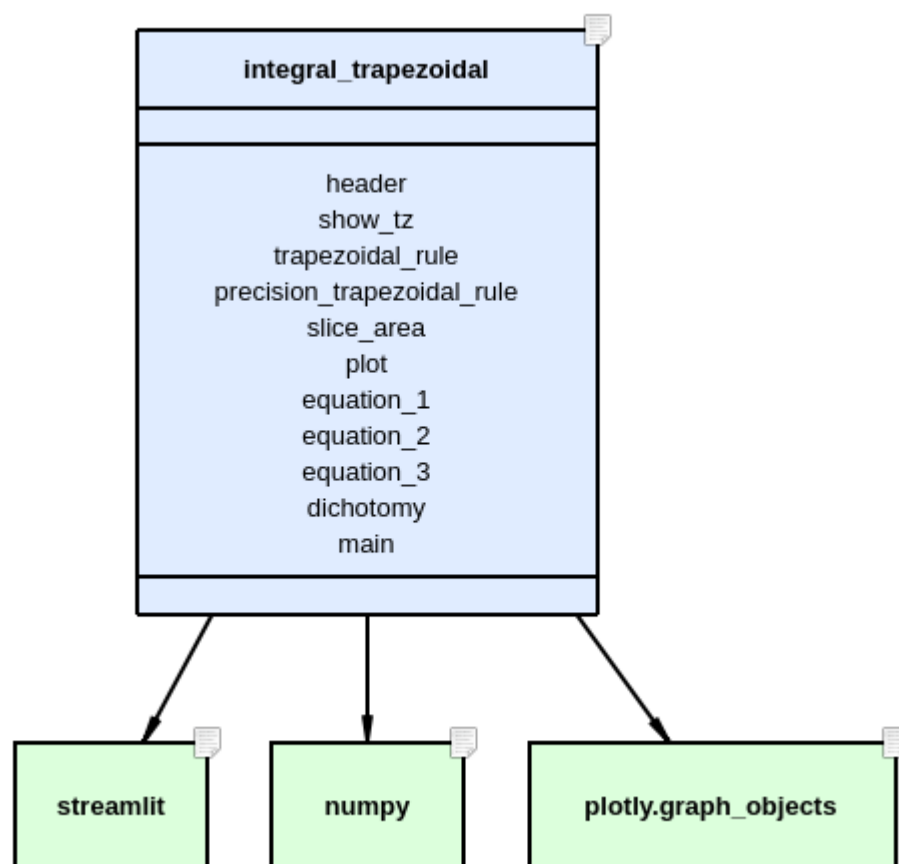
Программный продукт не предъявляет требований к безопасности системы.

- Требования на интеллектуальную собственность

Программный продукт не обладает правами на интеллектуальную собственность.

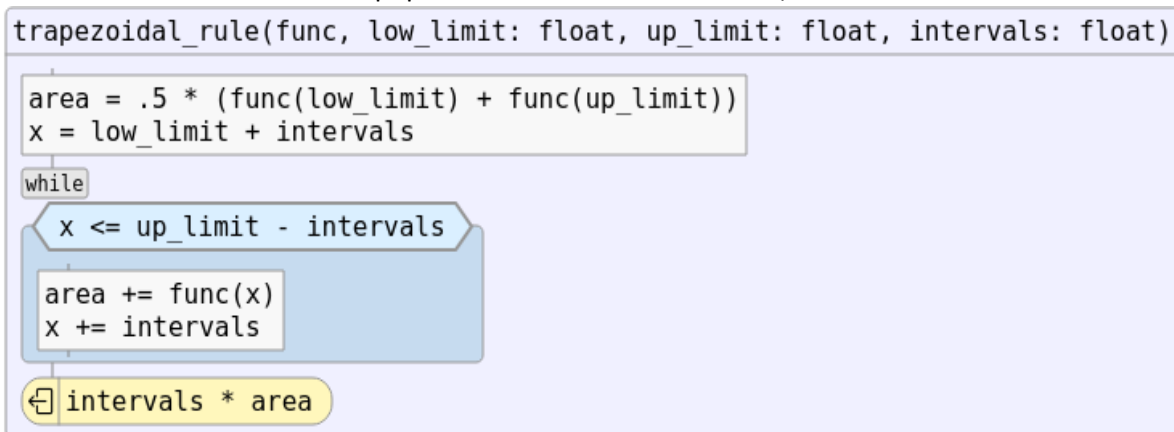
Блок-схемы алгоритмов основных функций системы

Верхнеуровневая диаграмма сущностей и подключаемых библиотек

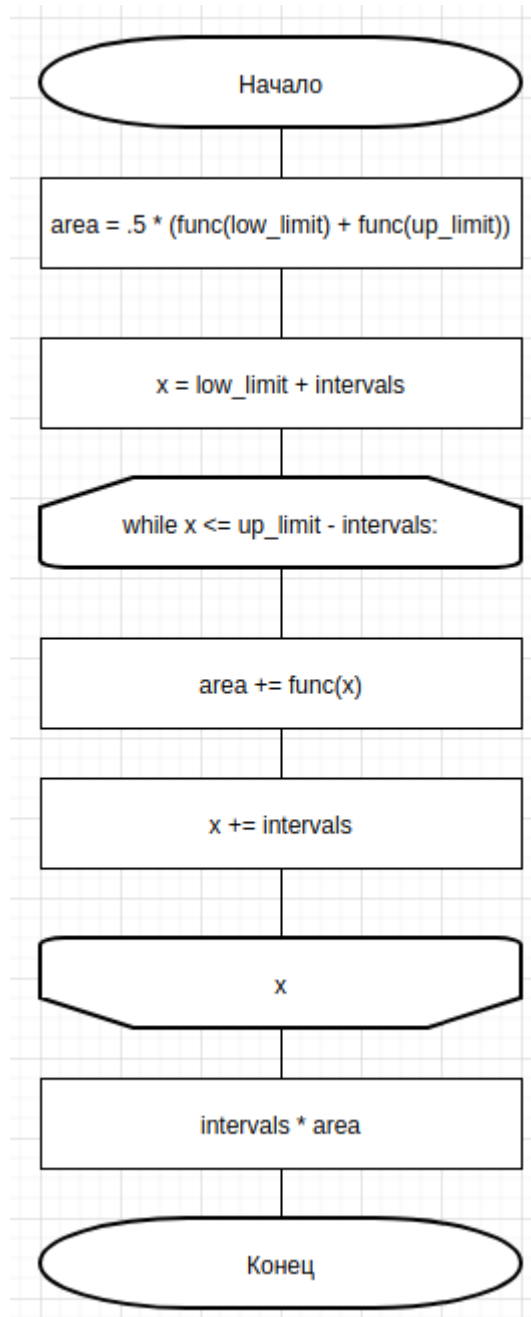


Блок схемы функции `trapezoidal_rule`:

Автоматически сгенерированная блок схема с помощью Codimension:

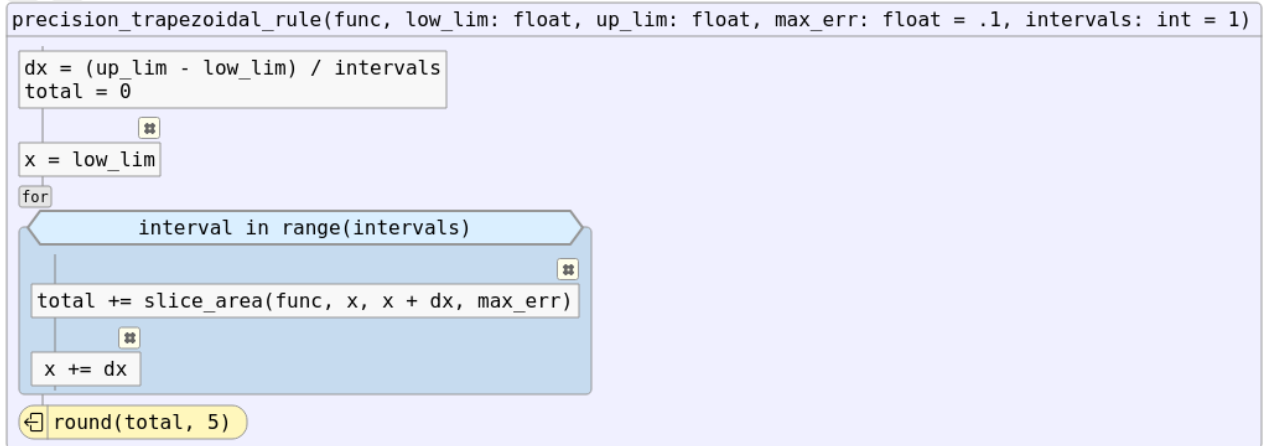


Классическая блок-схема (ГОСТ):

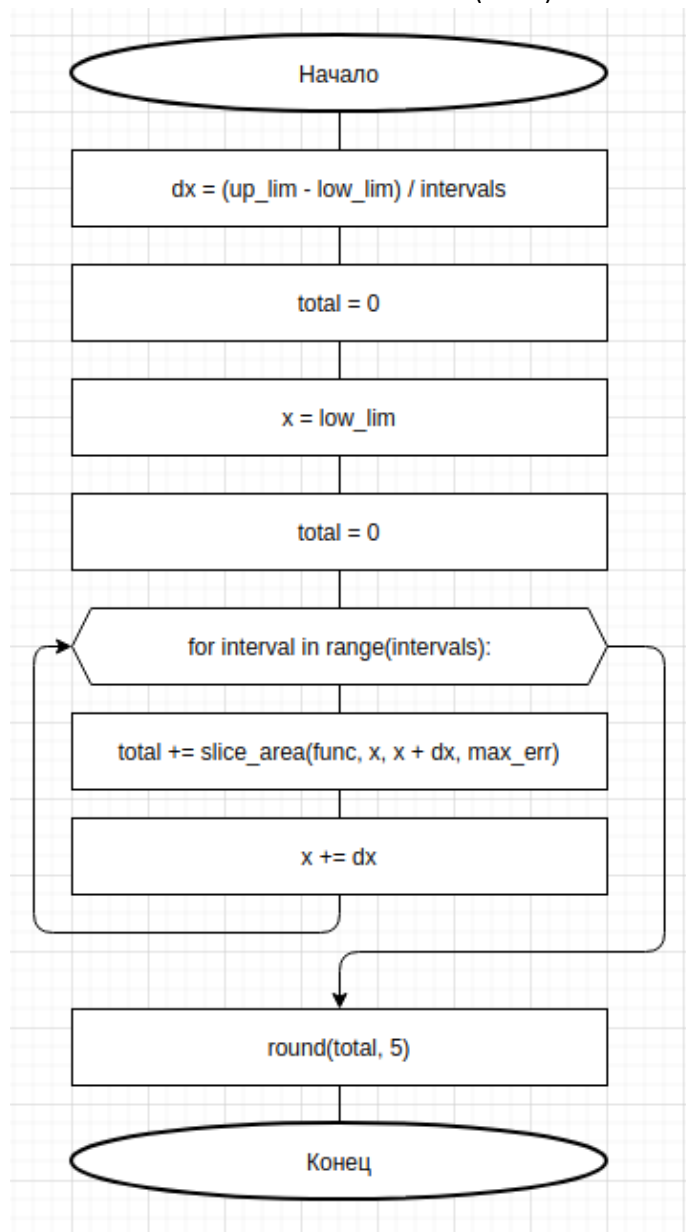


Блок схема функции precision_trapezoidal_rule:

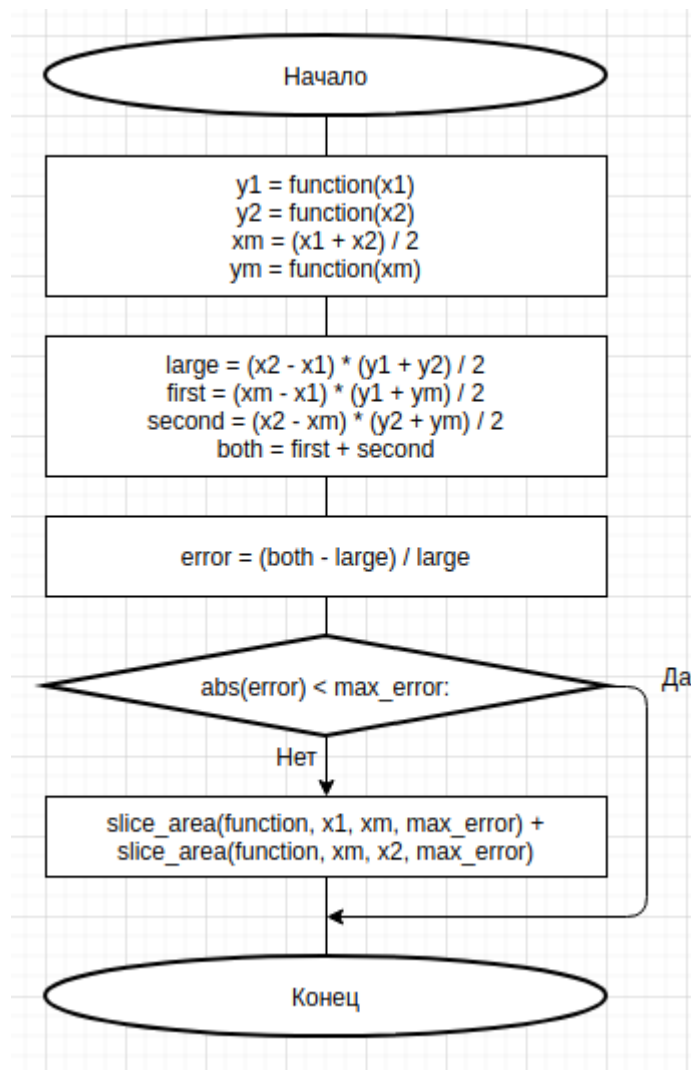
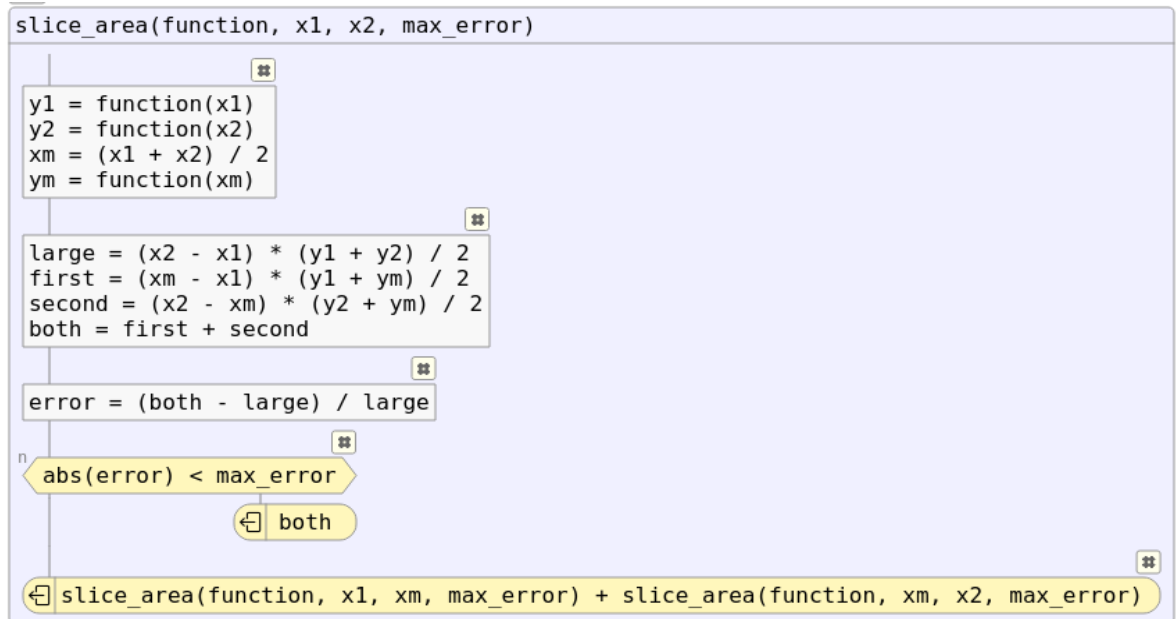
Автоматически сгенерированная блок-схема с помощью Codimension:



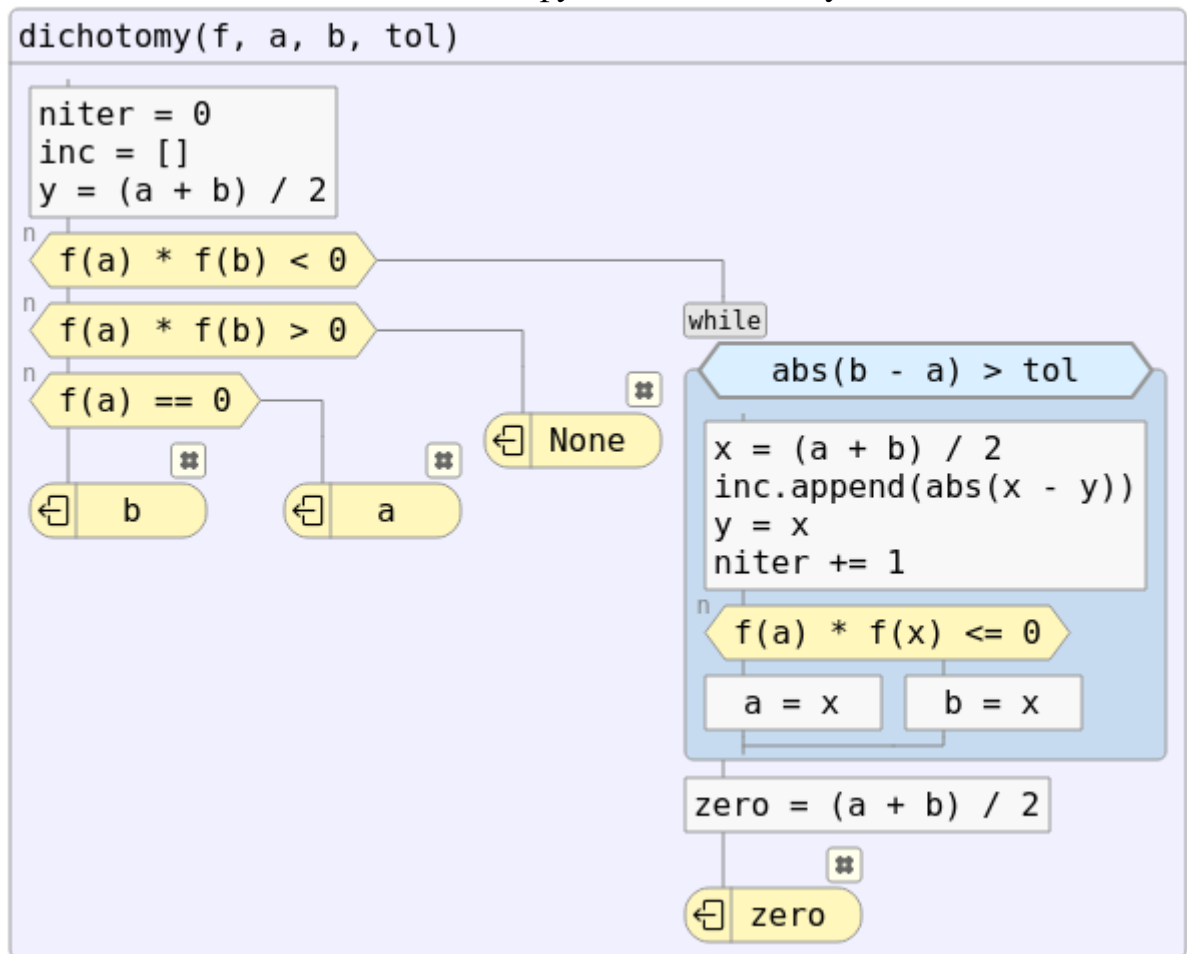
Классическая блок-схема (ГОСТ):

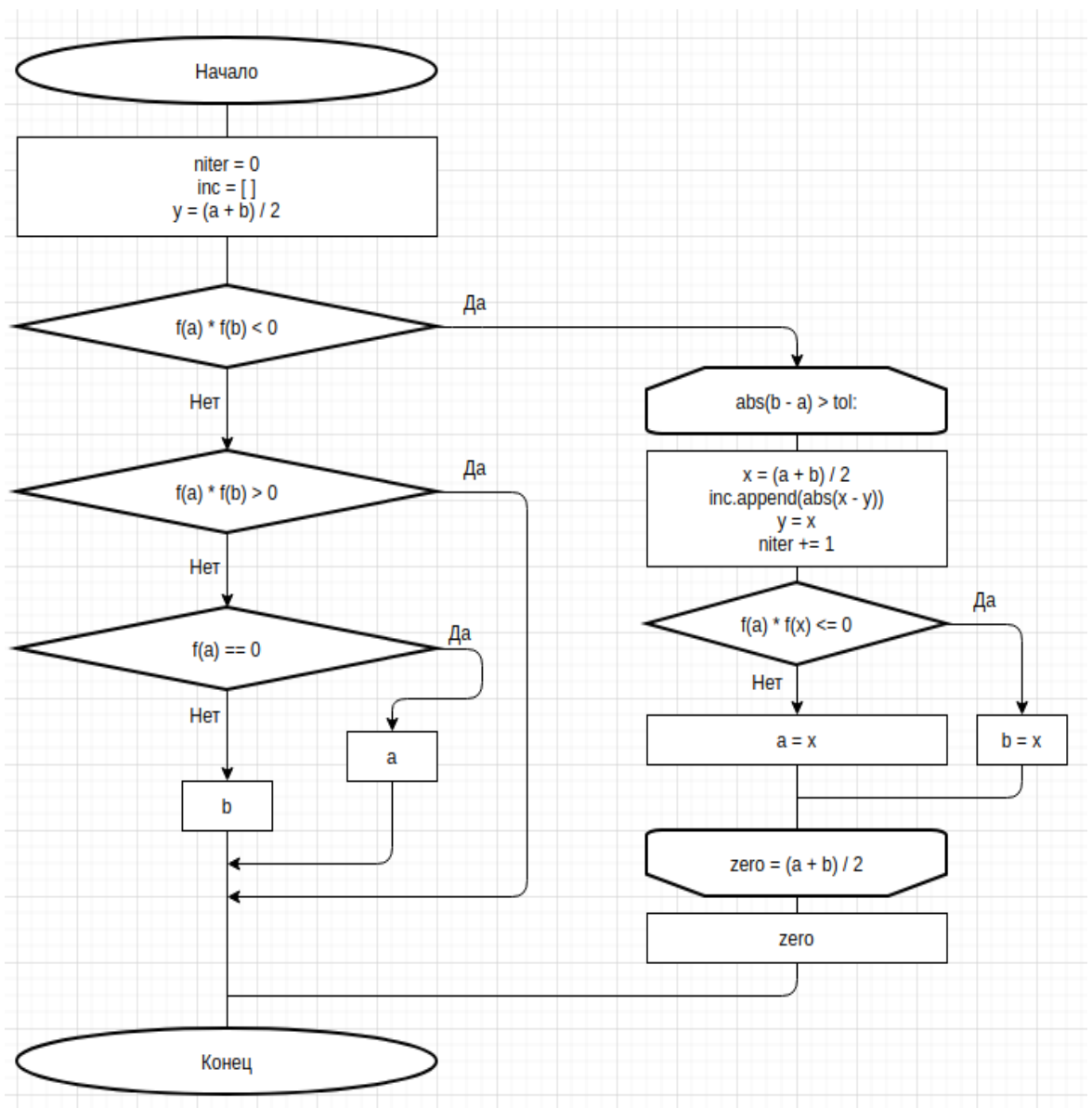


Блок схема вспомогательный функции slice_area для функции precision_trapezoidal_rule:



Блок схема функции dichotomy:





Код основных функций программы

Основные функции программы:

Функция для приближенного вычисления определенного интеграла методами трапеций с заданным шагом:

```
def trapezoidal_rule(func, low_limit: float, up_limit: float, intervals: float) -> float:
    """
        Правило трапеций для численной аппроксимации интегральной заданной функции
        :param func: математическая функция
        :param low_limit: нижний предел интегрирования
        :param up_limit: верхний предел интегрирования
        :param intervals: число отрезков, на которые разбивается
        :return: результат вычисления
    """
    area = .5 * (func(low_limit) + func(up_limit))
    x = low_limit + intervals
    while x <= up_limit - intervals:
        area += func(x)
        x += intervals
    return intervals * area
```

Функция для приближенного вычисления определенного интеграла методами трапеций с заданной точностью:

```
def precision_trapezoidal_rule(func, low_lim: float, up_lim: float, max_err: float = .1, intervals: int = 1) -> float:
    """
        Правило трапеций с заданной точностью
        :param func: математическая функция
        :param low_lim: нижний предел интегрирования
        :param up_lim: верхний предел интегрирования
        :param max_err: заданная точность
        :param intervals: число отрезков, на которые разбивается
        :return: результат вычисления
    """
    dx = (up_lim - low_lim) / intervals
    total = 0

    # выполняем интеграцию
    x = low_lim
    for interval in range(intervals):
        # добавляем область трапеции для этого среза
        total += slice_area(func, x, x + dx, max_err)

        # переходим к следующему срезу
        x += dx

    return round(total, 5)
```

Вспомогательная функция для приближенного вычисления определенного интеграла методами трапеций с заданной точностью, вызываемая внутри функции `precision_trapezoidal_rule`:

```
def slice_area(function, x1, x2, max_error):  
    # вычисляем функцию в конечных и средних точках  
    y1 = function(x1)  
    y2 = function(x2)  
    xm = (x1 + x2) / 2  
    ym = function(xm)  
  
    # рассчитываем площади срезов и самого большого участка  
    large = (x2 - x1) * (y1 + y2) / 2  
    first = (xm - x1) * (y1 + ym) / 2  
    second = (x2 - xm) * (y2 + ym) / 2  
    both = first + second  
  
    # рассчитываем ошибку  
    error = (both - large) / large  
  
    # сравниваем ошибку с допустимым значением ошибки (точности)  
    if abs(error) < max_error:  
        return both  
  
    # если ошибка больше допустимого значения - делим ее на две части (два среза)  
    return slice_area(function, x1, xm, max_error) + slice_area(function, xm, x2, max_error)
```

Функция для вычисления точки пересечения выбранной пользователем функции с осью абсцисс на заданном интервале методом дихотомии:

```
def dichotomy(f, a, b, tol):
    niter = 0
    inc = []
    y = (a + b) / 2

    if f(a) * f(b) < 0:
        while abs(b - a) > tol:
            x = (a + b) / 2
            inc.append(abs(x - y))
            y = x
            niter += 1
            if f(a) * f(x) <= 0:
                b = x
            else:
                a = x
        zero = (a + b) / 2
        st.write(f"Функция {f.__doc__} пересекает ось абсцисс в точке [{round(zero, 2)}, 0]")
        return zero
    elif f(a) * f(b) > 0:
        st.error(f"Невозможно применить метод дихотомии для функции {f.__doc__} на интервале [{a}, {b}]")

    else:
        if f(a) == 0:
            st.write(f"Ноль функции {f.__doc__} на [{a}, {b}] над точкой [{round(a, 2)}, 0]")
            return a
        else:
            st.write(f"Ноль функции {f.__doc__} на [{a}, {b}] b = [{round(b, 2)}, 0]")
            return b
```


Результаты тестирования

При запуске и тестировании программы были получены следующие результаты:

Выберите необходимое вычисление

- ☒ 1. Приближенное вычисление определенного интеграла методом трапеций с заданным шагом
- ☐ 2. Приближенное вычисление определенного интеграла методом трапеций с заданной точностью
- ☐ 3. Вычисление точки пересечения функции с осью абсцисс на заданном интервале методом дихотомии

Приближенное вычисление определенного интеграла методом трапеций с заданным шагом

Введите нижний предел: - + - +

Результат для $\sin(x)$ = 0.9892

Результат для $\cos^2(x) * \ln^2(x+5)$ = 2.26454

- ☒ 2. Приближенное вычисление определенного интеграла методом трапеций с заданной точностью
- ☐ 3. Вычисление точки пересечения функции с осью абсцисс на заданном интервале методом дихотомии

Приближенное вычисление определенного интеграла методом трапеций с заданной точностью

Введите нижний предел: - + - +

Результат для $\sin(x)$ = 0.98634

Результат для $\cos^2(x) * \ln^2(x+5)$ = 2.22693



3. Вычисление точки пересечения функции с осью абсцисс на заданном интервале методом дихотомии

Вычисление точки пересечения функции с осью абсцисс на заданном интервале методом дихотомии

Начало интервала (a):

-5.00

-

+

Конец интервала (b):

12.50

-

+

Эпсилон (e):

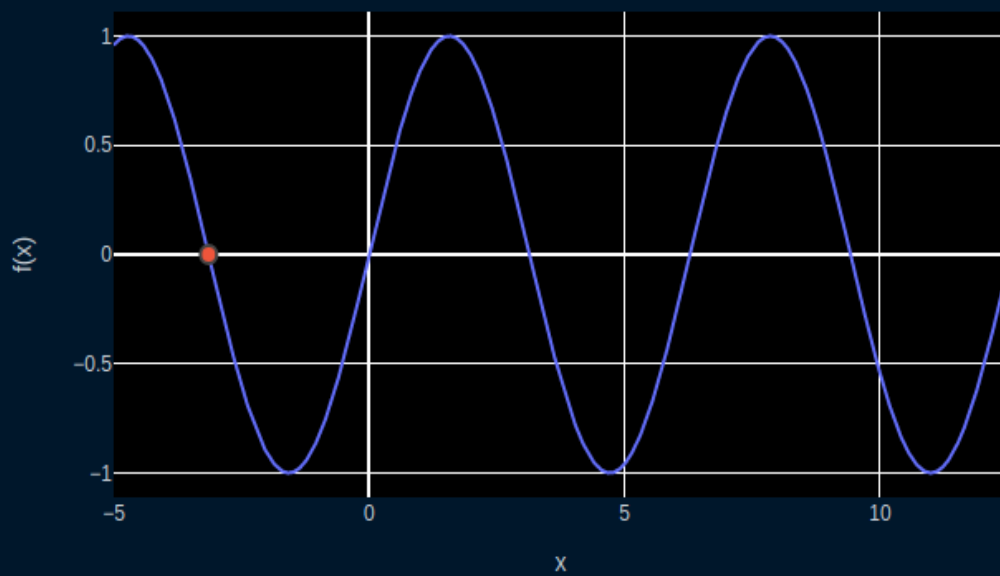
0.00001000

-

+

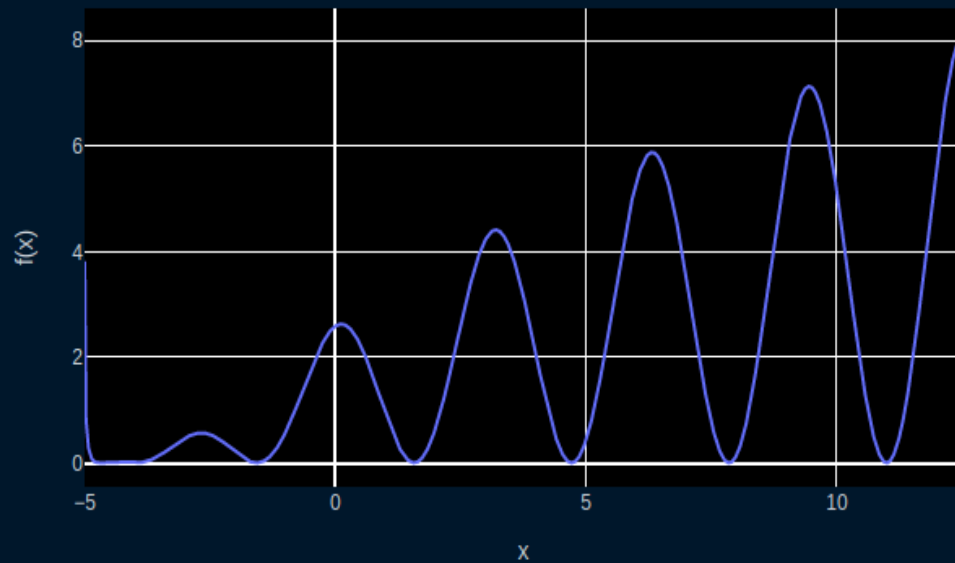
Функция $\sin(x)$ пересекает ось абсцисс в точке $[-3.14, 0]$

Функция $\sin(x)$



Невозможно применить метод дихотомии для функции $\cos^2(x) * \ln^2(x+5)$ на интервале $[-5.0, 12.5]$

Функция $\cos^2(x) * \ln^2(x+5)$



Результат

В результате работы была достигнута поставленная цель:

1. Создана программа в соответствии с описанным заданием
2. Реализованы 3 варианта вычислений
3. Реализована возможность параметризации вычислений пользователем
4. Реализовано отображение графиков и их изменений в зависимости от заданных параметров

Код программы находится в открытом репозитории по ссылке:

https://github.com/dKosarevsky/SEF_lab

Работающую программу можно увидеть и протестировать по ссылке:

https://share.streamlit.io/dkosarevsky/sef_lab_001/main/integral_trapezoidal.py

Программа написана на языке программирования Python 3.8.8 с использованием следующих библиотек:

- streamlit
- numpy
- plotly