Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Новосибирский государственный технический университет»

Кафедра Автоматизированных электроэнергетических систем

**Отчет ПО ПРАКТИКЕ**

Учебная практика: практика по получению первичных навыков работы с программным обеспечением

(наименование практики в соответствии с учебным планом)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Направление подготовки: **\_\_\_**«Электроэнергетика и электротехника»

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |  |
| --- | --- |
| Выполнил:  Студент Полозов А. А.  (Ф.И.О.)  Группа ЭН2-31  Факультет ФЭН.  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  подпись  «01» 06 2024 г. | Проверил:  Руководитель от НГТУ \_\_Дулов И.В.\_\_\_\_\_\_\_  (Ф.И.О.)  Балл: \_\_\_\_\_\_\_\_\_\_, ECTS\_\_\_\_\_\_\_\_\_\_\_\_\_,  Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  «отлично», «хорошо», «удовлетворительно», «неуд.»  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  подпись  «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2024 г. |

Новосибирск 2024

Оглавление

[1. Цель работы 3](#_Toc167442966)

[2. Задание 3](#_Toc167442967)

[3. Описание исходных данных 3](#_Toc167442968)

[4. Описание языка программирования Python 4](#_Toc167442969)

[5. Пояснения к заданию 5](#_Toc167442970)

[5.1. Математическая модель 5](#_Toc167442971)

[5.2. Численное интегрирование при помощи встроенной функции 6](#_Toc167442972)

[5.3. Численное интегрирование при помощи метода левых прямоугольников, реализованным самостоятельно 8](#_Toc167442973)

[Список используемых источников 10](#_Toc167442974)

# 1. Цель работы

Получение навыков алгоритмизации, разработки и тестирования программ на языке Python на примере задачи из области электроэнергетики.

# 2. Задание

1. Ознакомиться с поставленной задачей.
2. Выбрать подходящую среду разработки.
3. Определить библиотеки для решения задачи.
4. Составить математическую модель решения задачи с описанием используемых библиотек и функций. Каждый вариант должен быть решён двумя способами:
   1. Численное интегрирование при помощи встроенной функции.
   2. Численное интегрирование при помощи метода левых/правых прямоугольников, реализованным самостоятельно.
5. Составить алгоритм решения задачи в графическом виде.
6. Разработать интерфейс программы, содержащий следующий набор элементов:
   1. Поля ввода, позволяющие вводить исходные данные.
   2. Выпадающий список для выбора способа решения (реализованная или встроенная функции).
   3. Кнопка для запуска программы.
   4. Текст с результатами расчёта.
   5. Графики с необходимой графической информацией.
7. Выполнить реализацию составленного алгоритма в выбранной среде разработки.
8. Провести тестирование программы.

# 3. Описание исходных данных

Напряжение в электрической сети изменяется согласно выражению:

где:

– номинальное напряжение сети, В;

– частота тока, Гц;

– время, с.

Включённый в сеть вольтметр будет показывать значение, определяемое как:

где:

– период изменения напряжения, с.

Необходимо по показанию вольтметра определить номинальное напряжение сети. Отобразить графически вместе изменение напряжения во времени.

# 4. Описание языка программирования Python

Python — это высокоуровневый интерпретируемый язык программирования, который был разработан в конце 1980-х годов Гвидо ван Россумом в Нидерландах. Язык Python предоставляет простой и элегантный синтаксис, что делает его популярным средством для быстрого и эффективного написания программ на различных платформах.

### История

Python был первоначально создан как универсальный язык программирования, который мог бы облегчить задачи разработки. Со временем Python стал одним из самых популярных языков программирования, используемых в различных областях, таких как веб-разработка, научные исследования, анализ данных и машинное обучение.

### Типы и структуры данных

Python поддерживает различные типы данных, включая числа, строки, списки, кортежи, множества, словари и другие. Язык также предоставляет возможность создания пользовательских типов данных и структур.

### Синтаксис

Синтаксис Python легко читается и понимается, что делает его идеальным выбором для начинающих программистов. Python использует отступы для определения блоков кода, что способствует более читаемому и структурированному коду.

### Возможности

Python обладает мощными функциональностями, такими как динамическая типизация, автоматическое управление памятью, модульность, расширяемость и большое количество сторонних библиотек.

### Основные библиотеки

В Python существует огромное количество стандартных модулей и библиотек, таких как NumPy, Pandas, Matplotlib, TensorFlow, Flask и многие другие, которые обеспечивают расширенные возможности для решения различных задач.

### Применимость к практическим задачам

Python охватывает широкий спектр областей применения, включая веб-разработку, научные исследования, обработку данных, машинное обучение, искусственный интеллект, автоматизацию задач, разработку игр и многое другое. Язык Python позволяет быстро и эффективно решать разнообразные задачи.

### Применимость в энергетике

Python также широко применяется в энергетике для анализа данных, моделирования и прогнозирования, контроля и управления энергосистемами, разработки алгоритмов оптимизации и других задач. Python обладает богатым набором инструментов и библиотек для работы с временными рядами, машинным обучением, глубоким обучением и другими технологиями, которые могут быть полезны в энергетической отрасли.

# 5. Пояснения к заданию

## 5.1. Математическая модель

В уравнении является постоянной величиной, поэтому в уравнении её можно вынести за знак интеграла:

Если учесть, что , то уравнение примет вид:

Тогда номинальное напряжение можно найти следующим образом:

Если коэффициент пропорциональности обозначить за , то формула для расчёта номинального напряжения через показания вольтметра и частоту сети выглядит так:

где:

## 5.2. Численное интегрирование при помощи встроенной функции

Для нахождения определённого интеграла требуются следующие библиотеки:

1. numpy – для определения диапазонов определённых интегралов;
2. scipy – для вычисления численного решения интеграла.

Блок-схема алгоритма численного интегрирования при помощи встроенной функции представлена на рисунке 5.2.1.

|  |
| --- |
|  |
| Рис. 5.2.1 – Блок-схема алгоритма численного интегрирования при помощи встроенной функции |

Техническая реализация численного интегрирования при помощи встроенной функции представлена в листинге 5.2.1.

Данный программный код содержится в модуле common/built\_in\_integration.py.

### Листинг 5.2.1. – Техническая реализация численного интегрирования при помощи встроенной функции

import numpy as np

import scipy as sc

def alpha(f):

    p = np.pi

    integrand = lambda t: np.square(np.sin(2\*p\*f\*t) + 0.1 \* np.sin(6\*p\*f\*t))

    integral = sc.integrate.quad(integrand, 0, 1/f)

    result = np.power(f\*integral[0], -0.5)

    return result

## 5.3. Численное интегрирование при помощи метода левых прямоугольников, реализованным самостоятельно

**Метод прямоугольников** — метод численного интегрирования функции одной переменной, заключающийся в замене подынтегральной функции на константу, на каждом элементарном отрезке. Если рассмотреть график подынтегральной функции, то метод будет заключаться в приближённом вычислении площади под графиком суммированием площадей конечного числа прямоугольников, ширина которых будет определяться расстоянием между соответствующими соседними узлами интегрирования, а высота — значением подынтегральной функции в этих узлах.

Значение интеграла для элементарного участка находится по следующей формуле:

Блок-схема алгоритма численного интегрирования при помощи метода левых прямоугольников, реализованным самостоятельно, представлена на рисунке 5.3.1.

|  |
| --- |
|  |
| Рис. 5.3.1 – Блок-схема алгоритма численного интегрирования при помощи метода левых прямоугольников, реализованным самостоятельно |

Техническая реализация численного интегрирования при помощи метода левых прямоугольников, представлена в листинге 5.3.1.

Данный программный код содержится в модуле common/my\_integration.py.

### Листинг 5.3.1. – Техническая реализация численного интегрирования при помощи метода левых прямоугольников

import numpy as np

def alpha(f):

    p = np.pi

    integrand = lambda t: np.square(np.sin(2\*p\*f\*t) + 0.1 \* np.sin(6\*p\*f\*t))

    integral = integrate(integrand, 0, 1/f)

    result = np.power(f\*integral, -0.5)

    return result

def integrate(function, a, b):

    integral = 0

    x = a

    h = np.power(10.0, -5)

    while x < b:

        integral += function(x) \* h

        x += h

    return integral

# Список используемых источников

1. <https://www.geeksforgeeks.org/how-to-find-definite-integral-using-python/>
2. <https://ru.wikipedia.org/wiki/Метод_прямоугольников>