Министерство науки и высшего образования Российской Федерации Федеральное государственное бюджетное образовательное учреждение высшего образования

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)

Кафедра радиотехнических систем (РТС)

СРЕДСТВА ВИЗУАЛИЗАЦИИ ДАННЫХ НА ЯЗЫКЕ «PYTHON»

Отчет

По дисциплине

«Ознакомительная практика»

Студент гр. 123-3П

А. Н. Блинковский

2024 г.

Руководитель ст. преподаватель каф. РТС

Б.Ф. Ноздреватых

2024 г.

Томск 2023

Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение   
высшего образования

«Томский государственный университет систем управления и радиоэлектроники» (ТУСУР)

Кафедра радиотехнических систем (РТС)

УТВЕРЖДАЮ

Заведующий кафедрой РТС

\_\_\_\_\_\_\_\_\_\_ А.А. Мещеряков

29 августа 2024 г.

ЗАДАНИЕ

на учебную практику

Студенту, Блинковскому Алексею Николаевичу

Группа: 123-3П Факультет: РТФ

Тема работы: **СРЕДСТВА ВИЗУАЛИЗАЦИИ ДАННЫХ НА ЯЗЫКЕ PYTHON**

Целью данной работы является закрепление навыков разработки программного обеспечения на языке python и знакомство с основами визуализации данных программной инфраструктуры языка программирования.

1. Провести моделирование распределения мощности гармонической электромагнитной волны, излучаемой антенной базовой станции в точке приема мобильной станции на основе математической модели Окамура-Хата при следующих параметрах:

* диапазон рабочих частот: f = 150–1500 МГц;
* высота подъема антенны БС: ht = 30–200 м;
* высота подъема антенны МС: hr = 1–10 м;
* расстояние между БС и МС: r = 1–20 км.
  1. Моделирование провести в математических пакетах MathCAD и Matlab / Scilab, а также разработать программное обеспечение не языке python с графическим интерфейсом для отображения результатов моделирования.

1. Программа на python с графическим интерефейсом для построения высотно-временного распределения градиента температуры воздуха (Гамма,°С/100 м) в заданных интервалах времени от суток и меньше.
2. Визуализировать результаты экспериментальных данных в математических MathCAD и Matlab / Scilab, а также разработать программное обеспечение не языке python с графическим интерфейсом для отображения полученных результатов.
3. Сравнить результаты, полученные в математических программах и разработанном программном обеспечении.
4. Содержание отчета.
   1. Работа над индивидуальным заданием:

* теоретическая основа выполнения задания;
* решение индивидуального задания в MathCAD и Matlab / Scilab (листинг программы);
* наличие описания алгоритма (-ов) решения поставленной задачи (блок-схема);
* наличие оригинальных решений при разработке вопросов, поставленных заданием, либо возникших в ходе выполнения практики;
* наличие одного (или нескольких) тестовых примеров, показывающих точное выполнение решения поставленной задачи;
* - используемая литература;
* - выводы по полученным результатам (с точной адресацией к соответствующим материалам отчета), с формулировками предложений и рекомендаций, с обязательным сопоставлением пунктов задания с полученными результатами;
  1. Отчет должен быть выполнен в соответствии оформления текстовой, графической и программной документации предписаниям соответствующих стандартов (в том числе требованиям ОС ТУСУР);
  2. К отчету обязательно прилагается заполненный и оформленный по правилам дневник прохождения практики.

1. К отчету прилагается диск с электронной копией отчета и кодами программ.
2. Дата выдачи задания: 3 сентября 2024 г.
3. Срок сдачи на кафедру: 23 декабря 2024 г.

Руководитель практики: старший преподаватель каф. РТС

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Борис Федорович Ноздреватых

(подпись)

Студент:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Блинковский Алексей Николаевич

(подпись)

Оглавление

[Введение 5](#_TOC_250006)

1. [ТЕОРЕТИЧЕСКАЯ ЧАСТЬ 6](#_TOC_250005)
2. [ПРАКТИЧЕСКАЯ ЧАСТЬ 7](#_TOC_250004)
   1. [Расчет необходимых параметров и построение блок-схемы 7](#_TOC_250003)
   2. [Решение задачи разными способами 9](#_TOC_250002)

[Заключение 14](#_TOC_250001)

[Список использованных источников 15](#_TOC_250000)

Введение

Главная цель в этом курсовом проекте заключается в укреплении навыков разработки программного обеспечения на языке «Python» и ознакомлении с основами визуализации данных через программную инфраструктуру языка программирования. В рамках программы необходимо будет вычислить мощность гармонической электромагнитной волны, построить вертикальный метеорологический профиль в момент времени, заданный пользователем, а также вычислить разность температур между теоретической кривой и максимальным отклонением измеренной кривой и высоту температурного отклонения.

Программа должна также иметь графический интерфейс, позволяющий пользователю самостоятельно выбрать дату и время для наблюдения в течение дня и автоматически перестраивать график при изменении этих параметров. Помимо этого, необходимо проверить правильность выполнения задания в математических пакетах, таких как «Matlab» и «Mathcad». Мои вспомогательные и необходимые библиотеки включают «PySide2», «matplotlib» [1] и «Qt». В качестве исходных данных я использую данные MTP-5 за январь 2019 года.

1 ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

Метод Окамура-Хата описывает особенности распространения радиоволн над квази-плоской местностью и не учитывает особенности рельефа [5]. Распространение основных лучей от передающей станции происходит выше крыш строений (рисунок 1.1)

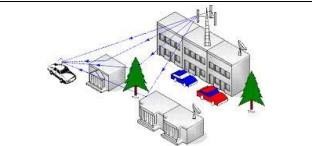


Рисунок 1.1 – Модель Окамура-Хата

Такая модель приближенно описывает зависимость мощности радиосигнала в точке приема от расстояния между БС и МС при заданных параметрах приемопередающих устройств и их антенных систем при различных условиях окружающей среды (рисунок 1.2)

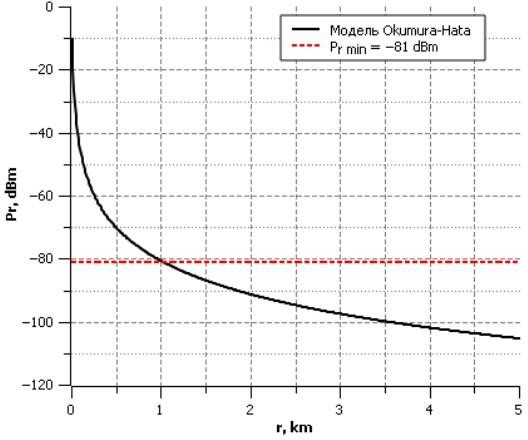


Рисунок 1.2 – Определение максимального радиуса соты по модели «Okumura-Hata» (𝑃𝑟 𝑚𝑖𝑛 = −81 дБм, 𝑅𝑚𝑎𝑥 = 1.021 км)

2 ПРАКТИЧЕСКАЯ ЧАСТЬ

* 1. Расчет необходимых параметров и построение блок-схемы

Для правильного и последовательного написания программы будем пользоваться определенным алгоритмом (или блок-схемой, изображенной на рисунке 2.1), чтобы выполнять все задачи по порядку, это поможет избежать сбоя или нежелательных проблем:

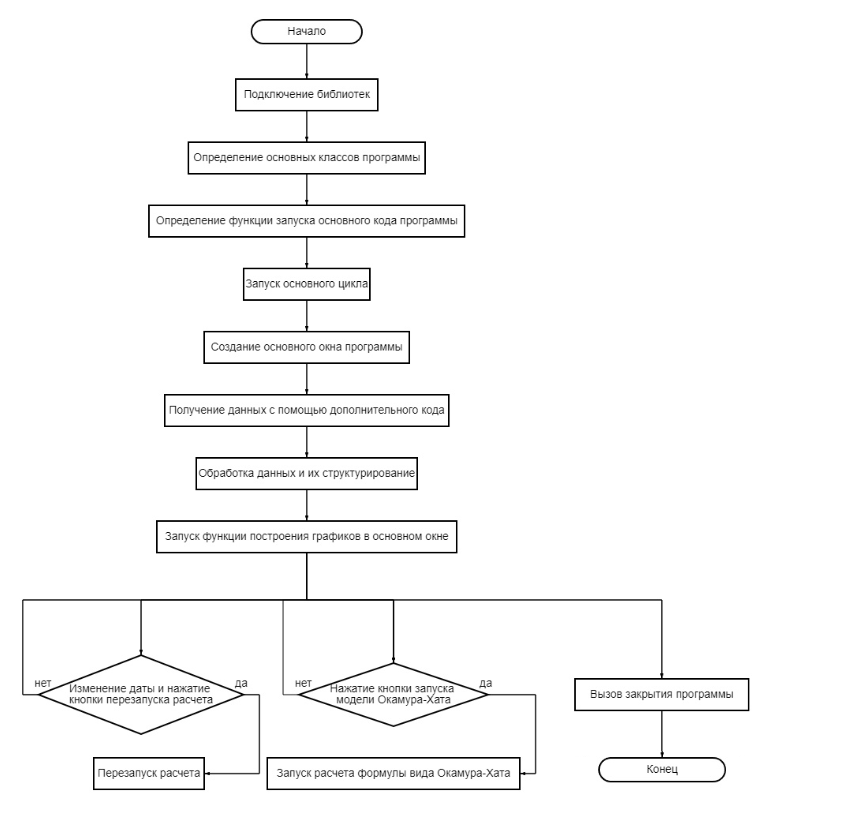


Рисунок 2.1 – Блок-схема

Метод Окамура-Хата содержится в файле «ohata.py».

Для выполнения расчетов в программе обратимся к библиотекам (рисунок 2.2):

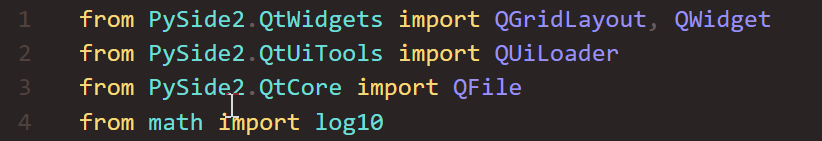


Рисунок 2.2 – Подключение библиотек в файле «ohata.py».

Ниже на рисунке (2.3) представлен код для решения по соответствующему методу:

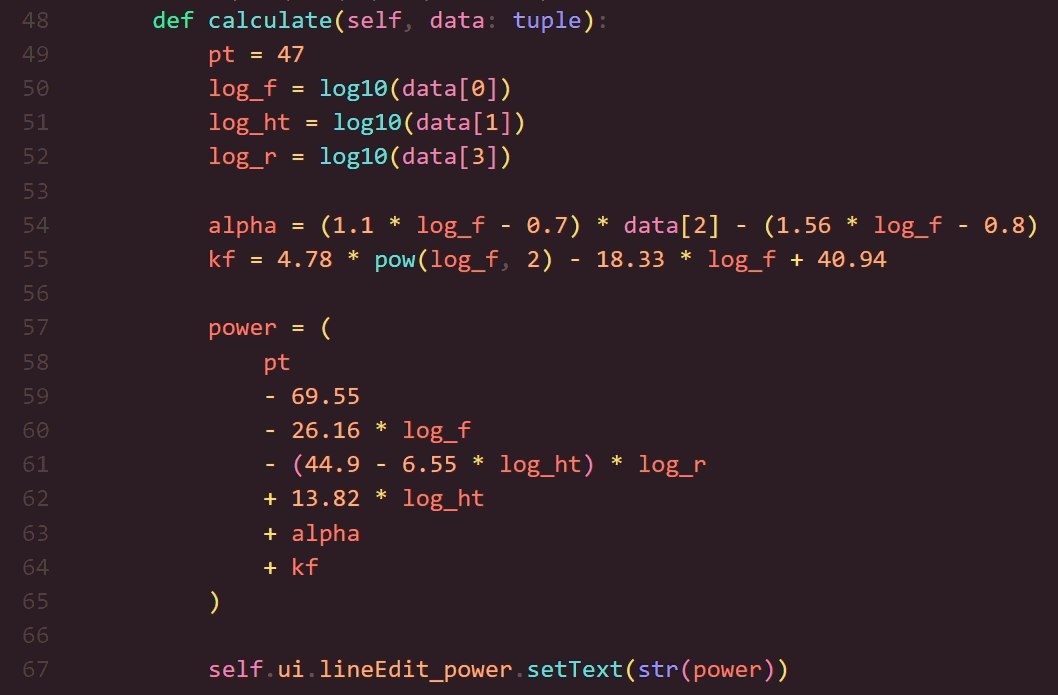


Рисунок 2.3 – Код для решения

Кортежи неизменяемые структуры данных. Так же как списки они могут состоять из элементов разных типов, перечисленных через запятую. Кортежи заключаются в круглые, а не квадратные скобки.

Мощность представлена на рисунке (2.4):

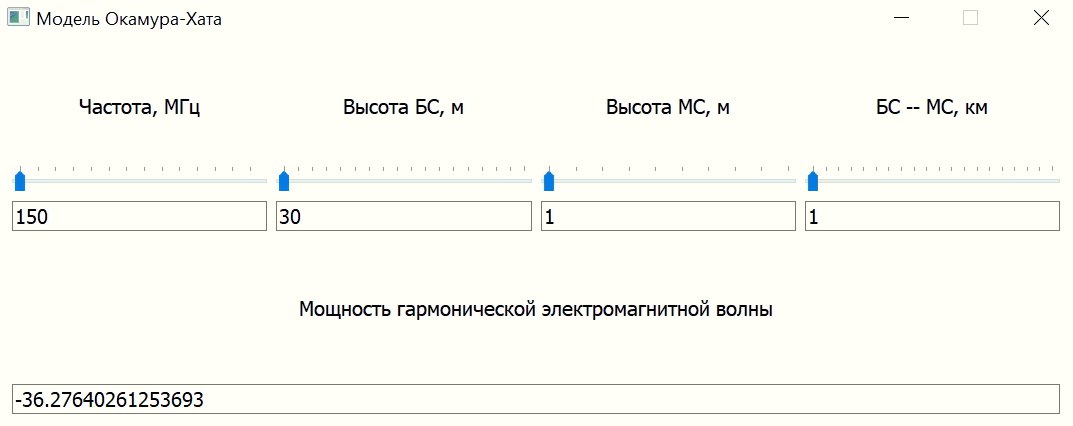


Рисунок 2.4 – Мощность, полученная при программировании на языке Python (в окне графического интерфейса)

Рассчитаем по методу Окамура-Хата необходимые параметры, теперь в математическом пакете MathCAD (рисунок 2.5):

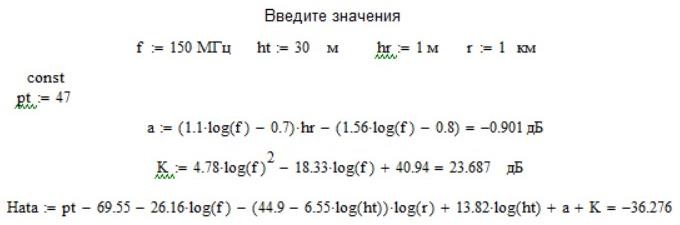


Рисунок 2.5 – метод Окамура-Хата в MathCAD

* 1. Решение задачи разными способами

Теперь произведем расчеты температуры на конкретной высоте по дате 1 февраля в момент времени 00 часов 00 минут, и построим график тремя разными способами: в математических пакетах MathCAD, Matlab и на языке Python. Первоначально, решим задачу и построим графике в MathCAD (рисунок 2.6):

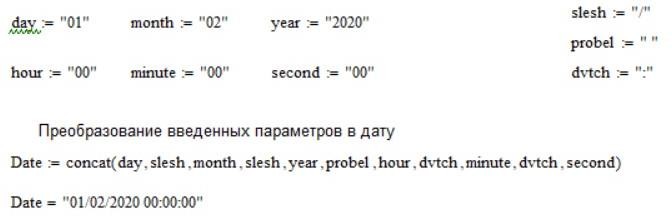


Рисунок 2.6 – Ввод параметров

Не забываем про парсинг данных (их много, они необходимы для построения графика). В конечном итоге получим вот такую таблицу (рисунок 2.7) и индексацию даты (рисунок 2.8):



Рисунок 2.7 – Таблица данных

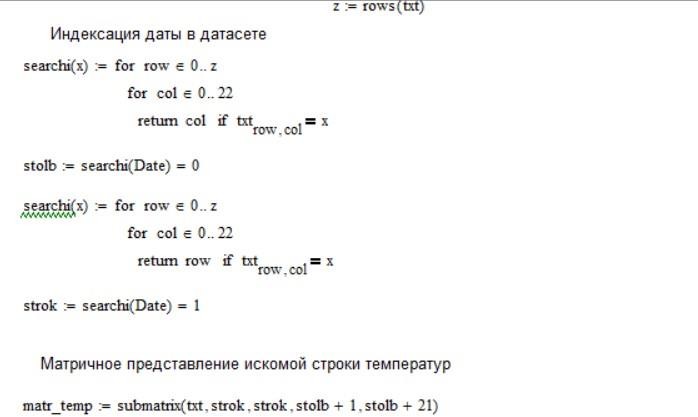


Рисунок 2.8 – Индексация даты и подключение матричного представления.

Получены массивы, представленные на рисунке (2.9):

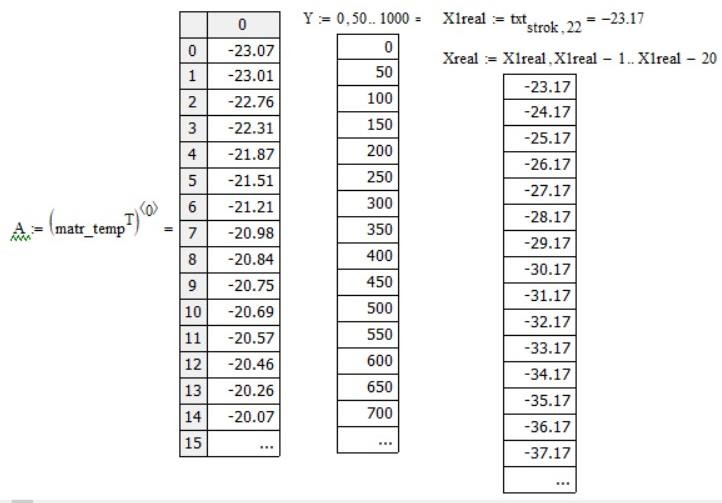


Рисунок 2.9 – Массивы как параметры осей графика Результатом будет график температур, представленный на рисунке

(2.10):

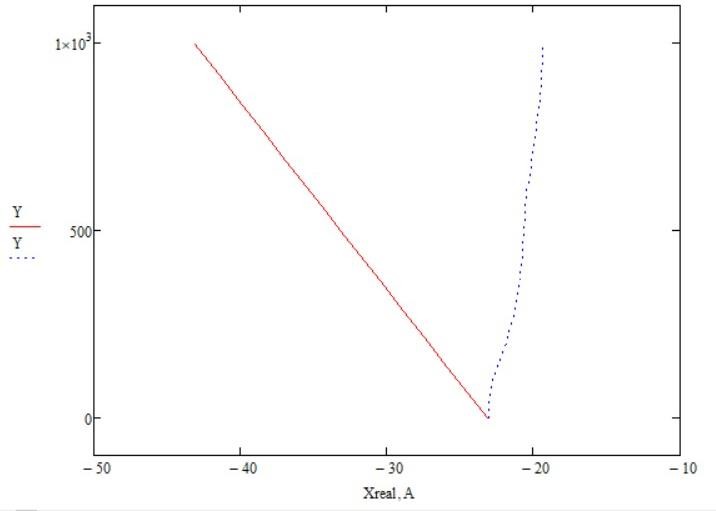


Рисунок 2.10 – График температур в MathCAD

Код для построения графика теоретической и фактической кривой температуры от высоты представлен на рисунке (2.11):



Рисунок 2.11 – Код для построения графика График температур представлен ниже (рисунок 2.12):

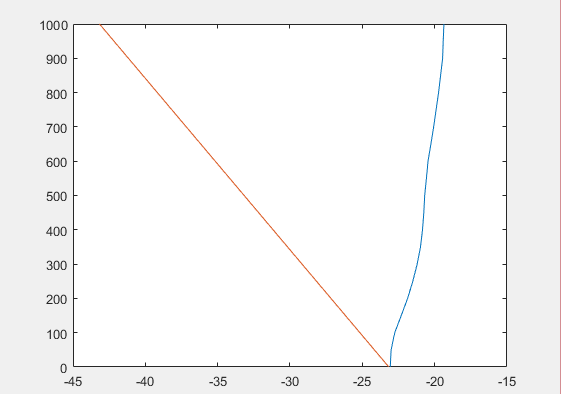


Рисунок 2.12 – График температур в Matlab Решим задачу и построим график на языке Python:

Данные несколько строк включает в себя масштабирование гарфика, построение осей, кривые «Теоретическая» и «Фактическая» (рисунок 2.13).

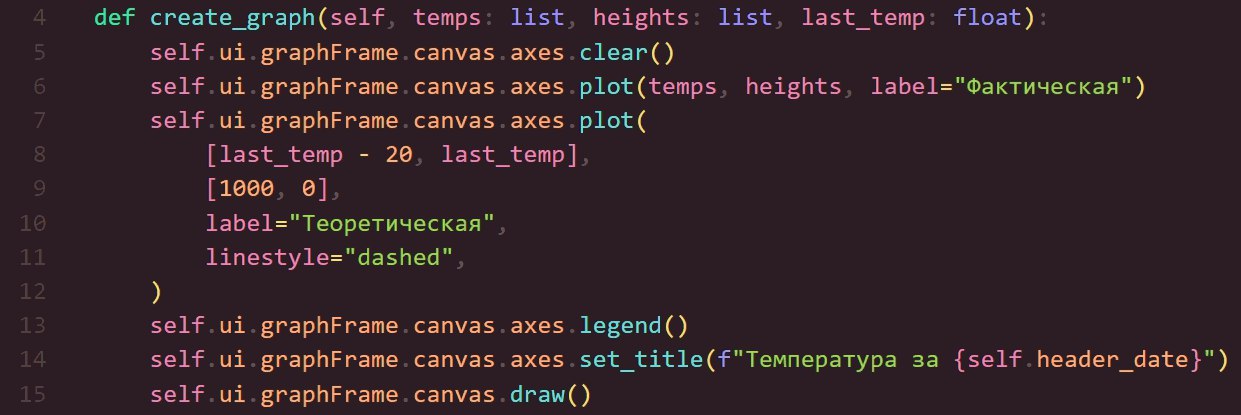


Рисунок 2.13 – Код для построения гарфика в Python [3] Получим график, выведенный в отдельном окне (рисунок 2.14):

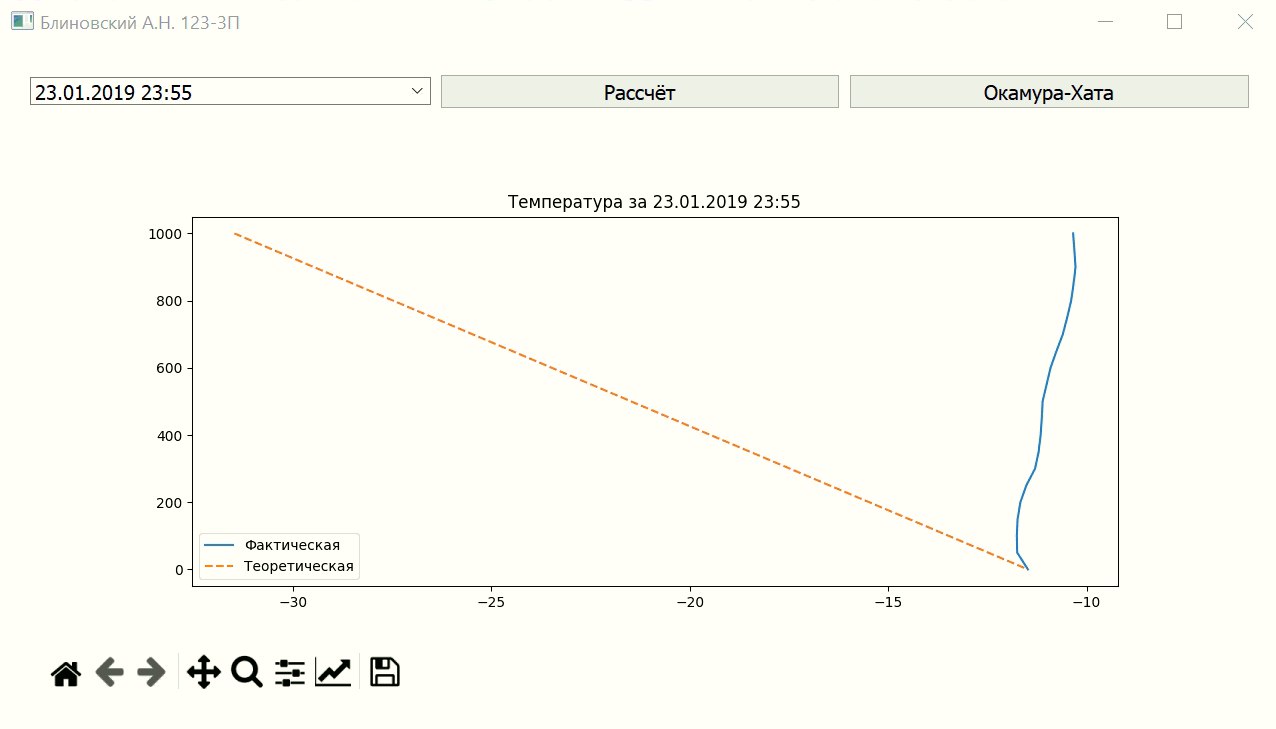


Рисунок 2.14 – График температуры, полученный при помощи программы

Посчитаем разницу температур между теоретической кривой с максимальным отклонением от измеренной кривой в течение месяца и посчитаем высоту, на которой находится это отклонение. Выполняются расчеты в MathCAD (рисунок 2.15), построение таблицы разностей (рисунок 2.16), и код для таблицы (рисунок 2.17):

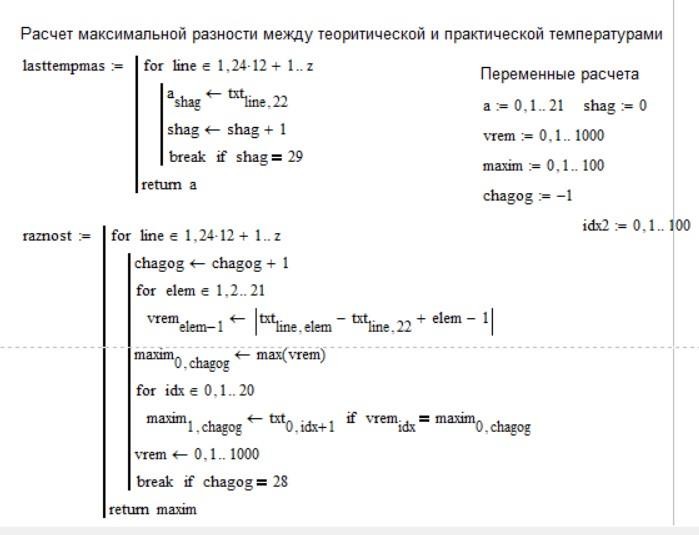


Рисунок 2.15 – Расчет разности между теоретической и практической температурами

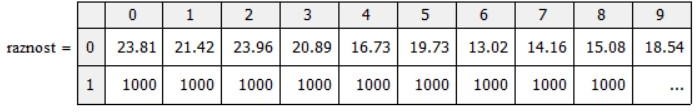


Рисунок 2.16 – Таблица разностей в MathCAD



Рисунок 2.17 – Таблица разностей в Python

Метод append() добавляет в конец списка элемент, переданный ему в качестве аргумента. Как и все методы в Python, он вызывается через оператор- точку [4].

Ниже представлен графический интерфейс таблицы (рисунок 2.18):



Рисунок 2.18 – Таблица разностей температур , написанная на языке Python [2]

Проанализировав полученные значения и графики, можно сделать вывод, что расчеты и построение выполнено правильно, а графики абсолютно идентичны друг другу.

Заключение

В данном курсовом проекте была проделана немалая и усердная работа: были применены знания и навыки, приобретенные в процессе обучения, а также доказаны на практике. В созданном приложении были соблюдены все необходимые требования, такие как: обязательное написание программы на языке Python и графический интерфейс; работоспособность программы протестирована на тестовых примерах, которые были придуманы самостоятельно; проверка и полное решение задания программно- выполняемых расчётов подтверждены в математических пакетах MathCAD и Matlab, где мы видим, что графики ничем друг от друга не отличаются и все вычисления выполнены верно.

Результат выполнения задания предоставлен в виде приложения под операционную систему Windows с графическим интерфейсом, для расчета мощности волны пользователь может менять все 4 параметра. Так же при построении графика пользователь может выбрать нужную ему дату и время.

Список использованных источников

1. Уроки по библиотеке Matplotlib [Электронный ресурс]. – Режим доступа: <https://devpractice.ru/matplotlib-lessons/>(Дата обращения: 2.11.2023)
2. Способ импортирования графического окна matplotlib [Электронный ресурс]. – Режим доступа: <https://www.programmersought.com/article/64688653319/> (Дата обращения: 3.11.2023)
3. Реализация вставки графика matplotlib в графический интерфейс [Электронный ресурс]. – Режим доступа: <https://www.youtube.com/watch?v=0qUV6mmSO3Q> (Дата обращения: 3.11.2023)
4. Как рассчитать стандартное отклонение списка в Python [Электронный ресурс]. – Режим доступа: <https://www.codecamp.ru/blog/standard-deviation-of-list-python/> (Дата обращения: 6.11.2023)
5. Статья про математические модели и алгоритмы распространения радиоволн в сотовых сетях мобильной связи [Электронный ресурс]. – Режим доступа: <https://sdo.tusur.ru/pluginfile.php/517341/mod_resource/content/1/Статья.pdf> (Дата обращения: 12.11.2023)