

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт перспективной инженерии
Департамент цифровых, робототехнических систем и электроники

ОТЧЕТ
ПО ЛАБОРАТОРНОЙ РАБОТЕ №1
дисциплины
«Основы теории электромагнитных полей и волн»

Выполнил:
Милькевич Александр Александрович
2 курс, группа ИТС-б-о-23-1,
11.03.02 «Инфокоммуникационные
технологии и системы связи,
направленность (профиль)
«Интеллектуальная обработка данных
в инфокоммуникационных системах и
сетях», очная форма обучения

(подпись)

Проверил:
Ассистент департамента цифровых,
робототехнических систем и
электроники Хацукова А.И

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2025 г.

Тема: Исследование поля элементарных излучателей

Цель: провести расчёт поля элементарного щелевого излучателя и поля элемента Гюйгенса. Провести их сравнение.

Ход выполнения работы:

1. Работа с ячейками Markdown. В соответствии с вариантом, повторить пример из методички по работе с ячейками Markdown в JupyterLab и Google Colab.

Практическое задание 1 ¶

Жирный и *курсивный* текст

Маркированный список:

- Первый
- Второй
- Третий

Нумерованный список:

1. Первый пункт
2. Второй пункт
3. Третий пункт

Формула согласно индивидуальному варианту:

$$A = \int_a^b |f(x) - g(x)| dx$$

Картинка:




Рисунок 1. Результат работы с ячейками Markdown в JupyterLab

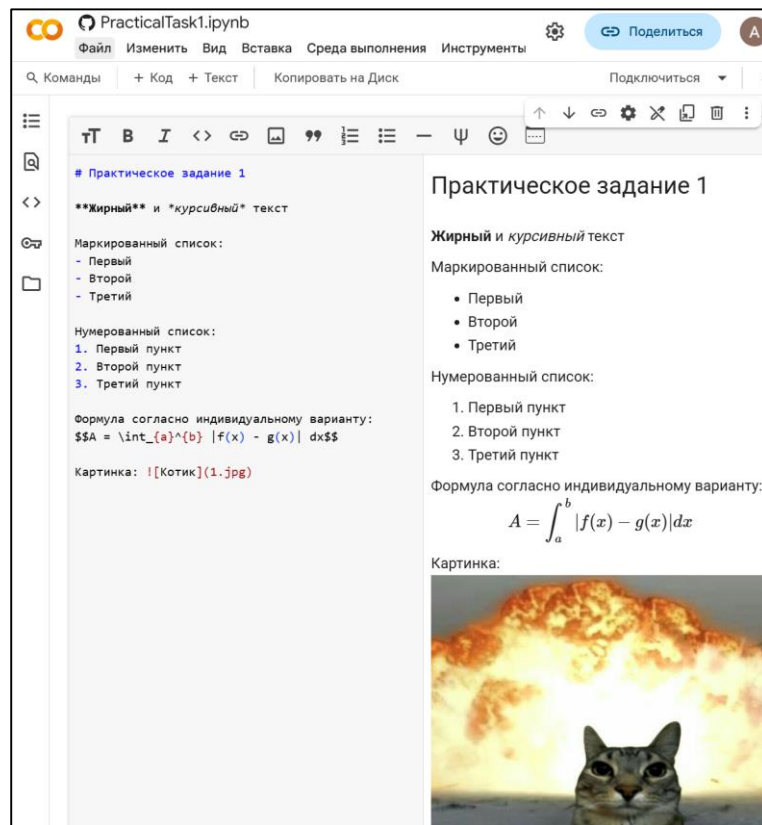


Рисунок 2. Результат работы с ячейками Markdown в Google Colab

2. Работа с файлами. Изучить загрузку, создание и сохранение файлов в Jupyter Notebook, JupyterLab и Google Colab.

```
import os

with open("test.txt", "w") as f:
    f.write("Первая строка\n")
    f.write("Вторая строка\n")

with open("test.txt", "r") as f:
    content = f.read()
    print(f"Содержимое файла: {content}")

Содержимое файла: Первая строка
Вторая строка

file_path="test.txt"
print(f"Файл: {os.path.exists(file_path)}")

Файл: True

os.remove("test.txt")
print(f"Файл : {os.path.exists(file_path)}")

Файл : False
```

Рисунок 3. Работа с файлами в JupyterLab



Рисунок 4. Работа с файлами в Jupyter Notebook



Рисунок 5. Работа с файлами в Google Colab

3. Магические команды. Изучить магические команды в Jupyter Notebook, JupyterLab и Google Colab.

```
[1]: %lsmagic

[1]: ▼ root:
    ► cell:
    ► line:

[2]: %env TEST = 5

env: TEST=5

[6]: %run Untitled.ipynb

12

i
ii
iii
iiii
iiiii
iiiiiii
Test while
Test while
Test while
Test while
Test while

2500
2000
1500
1000
500
0
0 10 20 30 40 50

[8]: %%time
import time
for i in range(50):
    time.sleep(0.1)

CPU times: total: 15.6 ms
Wall time: 5.13 s

[9]: %timeit x = [(i**10) for i in range(10)]

4.51 µs ± 269 ns per loop (mean ± std. dev. of 7 runs, 100,000 loops each)
```

Рисунок 6. Магические команды в JupyterLab

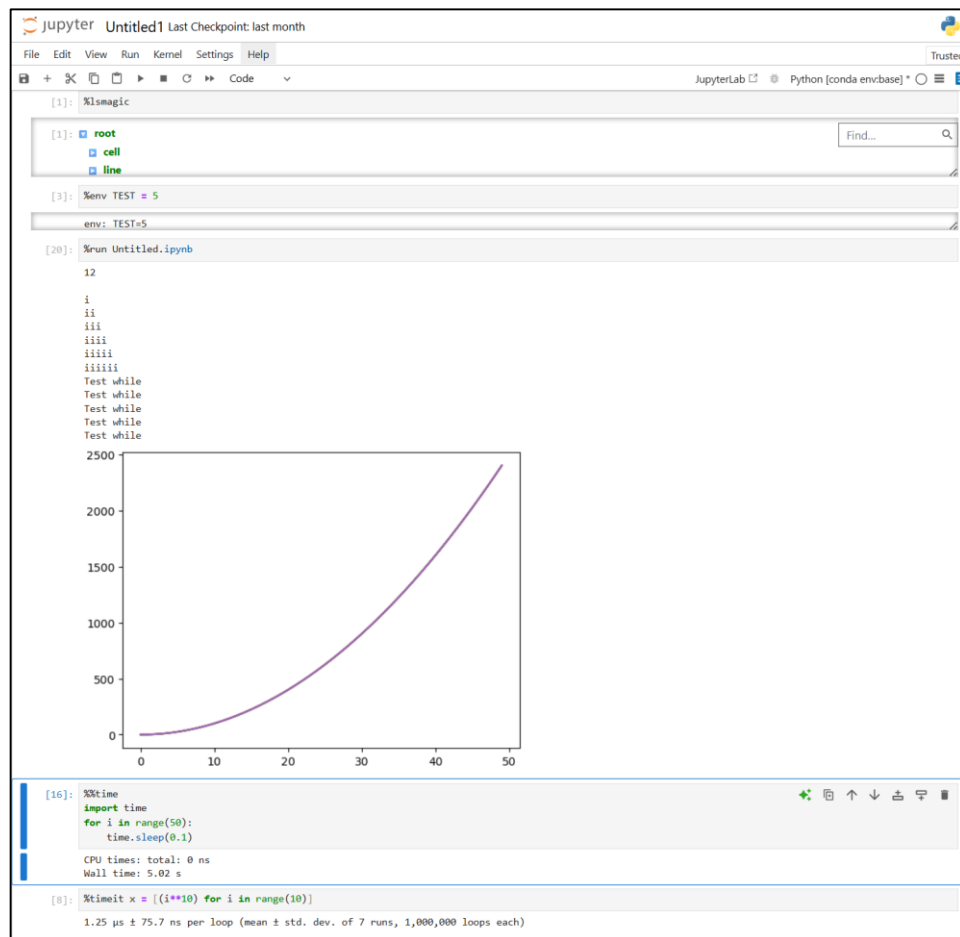


Рисунок 7. Работа с магическими командами в JupyterNotebook

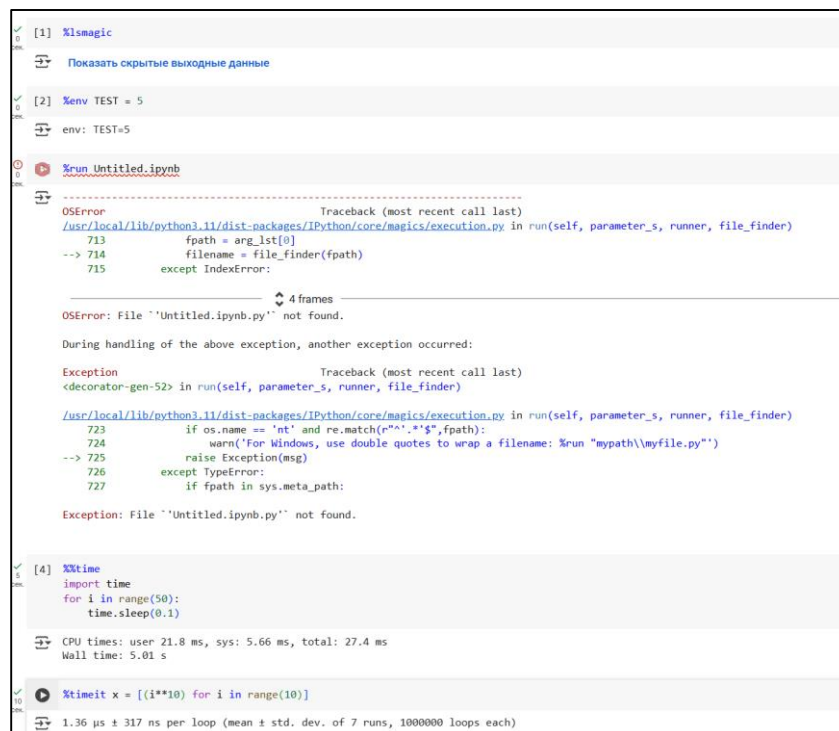


Рисунок 8. Работа с магическими командами в Google Colab

4. Взаимодействие с оболочкой системы. С вызвать выполнение команд терминала из JupyterLab и Google Colab.

```
!dir
!python --version
!mkdir test_folder
!dir
!rmdir test_folder
!dir

Том в устройстве C не имеет метки.
Серийный номер тома: 320C-B075

Содержимое папки C:\Users\student-09-302\Desktop\lab2\labii2\code

14.05.2025 14:31 <DIR> .
14.05.2025 14:31 <DIR> ..
14.05.2025 14:07 <DIR> .ipynb_checkpoints
14.05.2025 13:48 50 200 1.jpg
14.05.2025 13:55 2 002 PracticalTask1.ipynb
14.05.2025 13:58 2 842 Untitled.ipynb
14.05.2025 13:58 30 665 Untitled1.ipynb
14.05.2025 13:58 1 829 Untitled2.ipynb
14.05.2025 13:58 9 260 Untitled3.ipynb
14.05.2025 14:31 4 692 Untitled4.ipynb
14.05.2025 13:59 2 885 workWithFiles.ipynb
      8 файлов      104 375 байт
      3 папок    4 048 879 616 байт свободно
Python 3.11.4
Том в устройстве C не имеет метки.
Серийный номер тома: 320C-B075

Содержимое папки C:\Users\student-09-302\Desktop\lab2\labii2\code

14.05.2025 14:32 <DIR> .
14.05.2025 14:32 <DIR> ..
14.05.2025 14:07 <DIR> .ipynb_checkpoints
14.05.2025 13:48 50 200 1.jpg
14.05.2025 13:55 2 002 PracticalTask1.ipynb
14.05.2025 14:32 <DIR> test_folder
14.05.2025 13:58 2 842 Untitled.ipynb
14.05.2025 13:58 30 665 Untitled1.ipynb
14.05.2025 13:58 1 829 Untitled2.ipynb
14.05.2025 13:58 9 260 Untitled3.ipynb
14.05.2025 14:31 4 692 Untitled4.ipynb
14.05.2025 13:59 2 885 workWithFiles.ipynb
      8 файлов      104 375 байт
      4 папок    4 048 773 120 байт свободно
Том в устройстве C не имеет метки.
Серийный номер тома: 320C-B075

Содержимое папки C:\Users\student-09-302\Desktop\lab2\labii2\code

14.05.2025 14:32 <DIR> .
14.05.2025 14:32 <DIR> ..
14.05.2025 14:07 <DIR> .ipynb_checkpoints
14.05.2025 13:48 50 200 1.jpg
14.05.2025 13:55 2 002 PracticalTask1.ipynb
14.05.2025 13:58 2 842 Untitled.ipynb
14.05.2025 13:58 30 665 Untitled1.ipynb
14.05.2025 13:58 1 829 Untitled2.ipynb
14.05.2025 13:58 9 260 Untitled3.ipynb
14.05.2025 14:31 4 692 Untitled4.ipynb
14.05.2025 13:59 2 885 workWithFiles.ipynb
      8 файлов      104 375 байт
      3 папок    4 048 773 120 байт свободно
```

Рисунок 9. Работа с оболочкой JupyterLab

```
import os
print("Список файлов в текущей директории:")
!ls

Список файлов в текущей директории:
sample_data  test_folder

[ ] print("\nПуть к используемому Python:")
!which python

Путь к используемому Python:
/usr/local/bin/python

[ ] folder_name = "test_folder"

print(f"\nСоздание папки '{folder_name}':")
!mkdir {folder_name}

print("\nПроверка, что папка создана:")
!ls

Создание папки 'test_folder':

Проверка, что папка создана:
sample_data  test_folder

[ ] file_name = "test_file.txt"
file_path = os.path.join(".", file_name)
folder_path = os.path.join(".", folder_name)

#Создаем файл
if not os.path.exists(file_path):
    with open(file_path, "w") as f:
        f.write("Test content")
    print(f"Создан файл: {file_path}")

print(f"\nПеремещение файла '{file_name}' в папку '{folder_name}':")
!mv {file_name} {folder_name}

print("\nПроверка, что файл перемещен:")
!ls {folder_name}

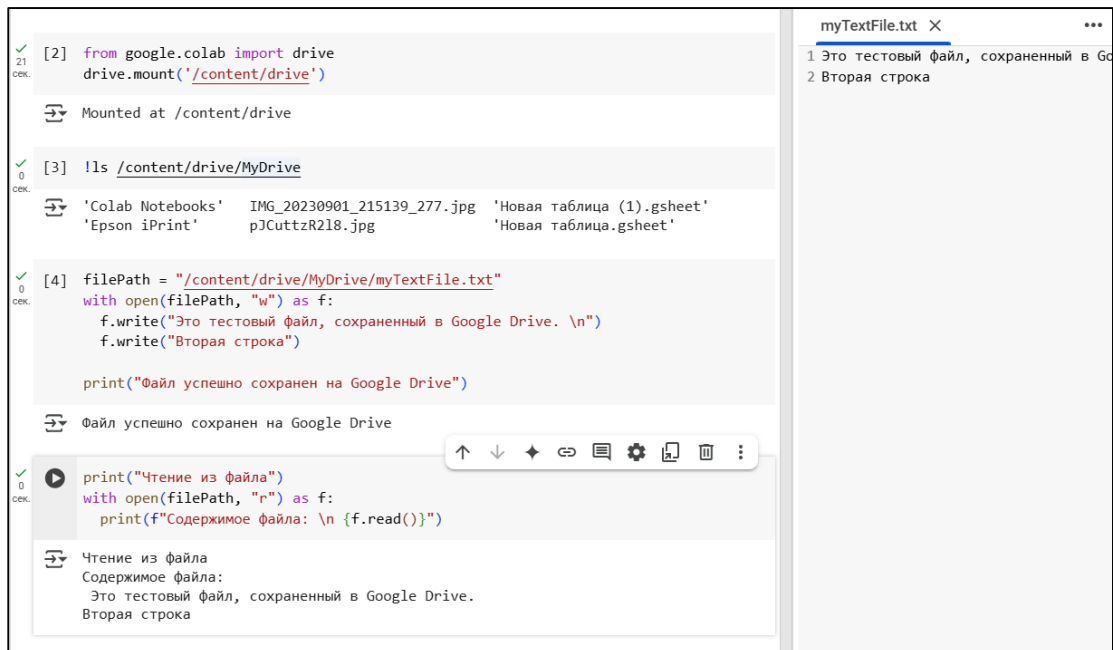
print(f"\nУдаление файла '{file_name}' из папки '{folder_name}':")
!rm {folder_path}/{file_name}

print("\nПроверка, что файл удален:")
!ls {folder_name}

Создан файл: ./test_file.txt
```

Рисунок 10. Работа с оболочкой в Google Colab

5. Работа с Google Drive в Google Colab. Подключить диск и создать на нём текстовый и csv файл.



The screenshot shows a Google Colab notebook with four code cells. The first cell mounts the Google Drive at `/content/drive`. The second cell lists the contents of `/content/drive/MyDrive`, showing files like `IMG_20230901_215139_277.jpg` and `Новая таблица (1).gsheet`. The third cell writes two lines of text to `myTextFile.txt` in the `MyDrive` directory. The fourth cell reads the content of `myTextFile.txt` and prints it. On the right, a preview of `myTextFile.txt` shows the two lines of text that were written.

```
[2] from google.colab import drive
drive.mount('/content/drive')

Mounted at /content/drive

[3] !ls /content/drive/MyDrive

'Colab Notebooks'  IMG_20230901_215139_277.jpg  'Новая таблица (1).gsheet'
'Epson iPrint'    pJCuttzR218.jpg             'Новая таблица.gsheet'

[4] filePath = "/content/drive/MyDrive/myTextFile.txt"
with open(filePath, "w") as f:
    f.write("Это тестовый файл, сохраненный в Google Drive. \n")
    f.write("Вторая строка")

print("Файл успешно сохранен на Google Drive")

Файл успешно сохранен на Google Drive

[5] print("Чтение из файла")
with open(filePath, "r") as f:
    print(f"Содержимое файла: \n {f.read()}")

Чтение из файла
Содержимое файла:
Это тестовый файл, сохраненный в Google Drive.
Вторая строка
```

Рисунок 11. Сохранение и чтение текстового файла



The screenshot shows a Google Colab notebook with a single code cell. The code creates a list of student records, each containing a name, age, and group. It then writes these records to a CSV file named `students.csv` in the `MyDrive` directory. On the right, a preview of `students.csv` shows the data written in a table format with three columns: `ФИО`, `возраст`, and `группа`.

```
[7] students = [
    ["ФИО", "возраст", "группа"],
    ["Милькевич Александр", 20, "ИТС-6-о-23-1"],
    ["Якушенко Антон", 19, "ИТС-6-о-23-1"],
    ["Пугачёв Кирилл", 20, "ИТС-6-о-23-1"]
]

pathForSCV = "/content/drive/MyDrive/students.csv"

with open(pathForSCV, "w") as file:
    for student in students:
        file.write(",".join(map(str, student)) + "\n")

print("Файл успешно создан")
```

ФИО	возраст	группа
Милькевич Александр	20	ИТС-6-о-23-1
Якушенко Антон	19	ИТС-6-о-23-1
Пугачёв Кирилл	20	ИТС-6-о-23-1

Рисунок 12. Создание csv файла

Вывод: в ходе выполнения лабораторной работы были изучены среды разработки Jupyter Notebook, JupyterLab и Google Colab. Также были изучены магические команды, облегчающие работу с дистрибутивом `anaconda` и подключение `google drive` к `google colab`.

Ссылка на гитхаб: <https://github.com/lolndo/labii2.git>

Ссылка на Google Drive: [Untitled1.ipynb - Colab](#)

Ответы на контрольные вопросы:

1. Основные отличия JupyterLab от Jupyter Notebook: JupyterLab представляет собой более продвинутую интегрированную среду разработки (IDE), которая объединяет функциональность Jupyter Notebook, текстового редактора, терминала и файлового менеджера в одном интерфейсе. В JupyterLab можно открывать и редактировать несколько файлов одновременно, работать с разными типами файлов (например, .ipynb, .py, .csv, .md), использовать встроенный терминал и изменять расположение панелей. Jupyter Notebook, напротив, ограничен линейной последовательностью ячеек и работой с отдельными тетрадями.

2. Чтобы создать новую рабочую среду (ноутбук) в JupyterLab, нужно в меню выбрать File => New => Notebook, затем выбрать доступное ядро (обычно Python). После этого откроется новая тетрадь, состоящая из ячеек.

3. В JupyterLab поддерживаются три типа ячеек: Code (для написания и выполнения программного кода), Markdown (для оформления пояснений и текста) и Raw (для хранения необработанного текста). Чтобы изменить тип ячейки, можно использовать горячие клавиши: Y для переключения на код, M для Markdown.

4. Чтобы выполнить код в ячейке, можно использовать горячие клавиши Shift+Enter (выполнить содержимое ячейки и перейти на ячейку ниже) или Alt+Enter (выполнить содержимое ячейки и вставить новую ячейку ниже).

5. Чтобы запустить терминал в JupyterLab, нужно выбрать File > New > Terminal. Текстовый редактор можно открыть через File > New > Text File.

6. В JupyterLab для работы с файлами и структурами каталогов используется файловый менеджер, который отображает структуру папок и файлов. Также можно использовать терминал для выполнения команд, связанных с файловой системой.

7. Управление ядрами (kernels) в JupyterLab осуществляется через меню Kernel. Можно перезапустить ядро, прервать выполнение кода или подключиться к другому ядру.

8. Основные возможности системы вкладок и окон в JupyterLab включают возможность открывать несколько файлов одновременно, изменять расположение панелей и работать с несколькими тетрадами в одном окне благодаря системе вкладок и разделению экрана.

9. В JupyterLab для измерения времени выполнения кода можно использовать магические команды `%timeit` и `%%time`. Пример: `%timeit sum(range(1000))` измеряет время выполнения команды `sum(range(1000))`.

10. Магические команды, такие как `%%bash`, `%%javascript`, `%%python`, позволяют запускать код на других языках программирования в JupyterLab. Например, `%%bash` позволяет выполнять команды оболочки Bash.

11. Основные отличия Google Colab от JupyterLab: Google Colab — это облачная среда, предоставляющая доступ к GPU и TPU бесплатно, интегрируется с Google Диском и позволяет работать с тетрадями Jupyter Notebook на удаленных серверах. JupyterLab — это локальная среда разработки с более широкими возможностями настройки и работы с файлами.

12. Чтобы создать новый ноутбук в Google Colab, нужно перейти в Файл => Новый ноутбук. Откроется рабочая область с первой ячейкой.

13. В Google Colab доступны два типа ячеек: Code (для написания и выполнения Python-кода) и Markdown (для оформления документации и текста). Переключение между типами ячеек осуществляется с помощью горячих клавиш `Ctrl+M Y` для кода и `Ctrl+M M` для Markdown.

14. Чтобы выполнить код в ячейке Google Colab, можно использовать горячие клавиши `Shift+Enter`. Также можно использовать `Alt+Enter` для выполнения кода и вставки новой ячейки ниже.

15. Google Colab поддерживает загрузку файлов через боковую панель (Файлы => Загрузить файлы) или с помощью команды `files.upload()`.

Сохранение файлов возможно на Google Диск или экспорт в форматы .ipynb, .py и другие.

16. Чтобы подключить Google Drive к Google Colab, нужно выполнить команду `from google.colab import drive` и затем `drive.mount('/content/drive')`. После этого можно работать с файлами на Google Диске.

17. Для загрузки файлов в Google Colab из локального компьютера используется команда `from google.colab import files` и `files.upload()`.

18. Чтобы посмотреть список файлов в среде Google Colab, можно использовать команду `!ls`.

19. В Google Colab для измерения времени выполнения кода используются магические команды `%timeit` и `%%time`. Пример: `%timeit sum(range(1000))` измеряет время выполнения команды `sum(range(1000))`.

20. Чтобы изменить аппаратные ресурсы в Google Colab (например, переключиться на GPU), нужно перейти в Среда выполнения => Изменить среду выполнения и в поле Аппаратный ускоритель выбрать GPU или TPU.