Вот объединённая шпаргалка по твоей программе, охватывающая:

1. 💡 Разницу между ОДУ и СДУ
2. ⚙️ Численные методы (Эйлер и Рунге-Кутта)
3. 📚 Используемые библиотеки
4. 🗜️ Как создаётся ZIP-файл с Excel
5. 🎨 Почему в Excel видны границы таблиц

**🧠 Шпаргалка по коду модели эпидемии**

**📌 1. ОДУ vs СДУ**

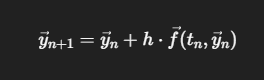
| **Характеристика** | **ОДУ** | **СДУ (в программе)** |
| --- | --- | --- |
| Описание | Одно уравнение | Система уравнений |
| Число переменных | Одна | Несколько (вектор) |
| Решение | Скалярная функция y(t) | Вектор-функция [S(t),I(t),...][S(t), I(t), ...] |
| Пример | Радиоактивный распад | Модель SIR, SEIR и т.п. |

В **вашем коде** используются **СДУ**, т.е. системы уравнений (например, sir\_model, seir\_model и др.)

**⚙️ 2. Численные методы**

**🔹 Метод Эйлера (наивный, но быстрый)**

Для СДУ:



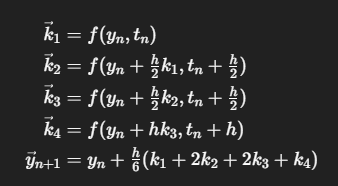
**В коде:**

dy = model\_func(y[i-1], t[i-1], \*args)

y[i] = y[i-1] + dy \* dt

**🔸 Метод Рунге-Кутта 4-го порядка (точнее)**

Для СДУ:



**В коде:**

k1 = model\_func(...)

...

y[i] = y[i-1] + (h/6)\*(k1 + 2\*k2 + 2\*k3 + k4)

**📚 3. Используемые библиотеки**

| **Библиотека** | **Назначение** |
| --- | --- |
| numpy (np) | Массивы, математика, интеграция |
| pandas (pd) | Хранение и экспорт результатов |
| matplotlib | Графики моделей |
| tkinter/ttk | GUI: окна, вкладки, кнопки |
| tkcalendar | Выбор даты |
| zipfile | Архивирование |
| BytesIO | Временное хранение файлов в памяти |

**🗜️ 4. Как создаётся ZIP с Excel-файлами**

🔧 В export\_results():

**Шаги:**

1. **Создаётся буфер в памяти:**
2. mem\_zip = BytesIO()
3. **Создаётся ZIP-файл в памяти:**
4. with zipfile.ZipFile(mem\_zip, mode='w', compression=zipfile.ZIP\_DEFLATED) as zf:
5. **Для каждой модели:**
   * создаётся Excel-файл в BytesIO
   * добавляются:
     + параметры (Параметры)
     + начальные данные (Начальные данные)
     + таблица с решением + метаинформация (Решение)
     + график (График)
   * записывается внутрь архива:
   * zf.writestr(f"{model\_name}.xlsx", excel\_buffer.getvalue())
6. **Сохраняется ZIP на диск:**
7. with open(save\_path, "wb") as f:
8. f.write(mem\_zip.getvalue())

**🎨 5. Почему в Excel видны границы таблиц?**

* Вы используете engine='xlsxwriter' в ExcelWriter, а он:
  + автоматически применяет стили и видимые рамки
  + добавляет графики, которые Excel отображает с таблицами с границами

В Python существует три ключевые библиотеки для работы с данными, научных вычислений и визуализации:

- NumPy — для численных операций и работы с многомерными массивами.

- pandas — для обработки и анализа табличных данных.

- Matplotlib — для построения графиков и визуализации данных.

1. NumPy (Numerical Python)

NumPy — это фундаментальная библиотека для научных вычислений в Python. Она предоставляет поддержку многомерных массивов (`ndarray`), математические функции для операций с ними, а также инструменты для интеграции с другими языками (C/C++, Fortran).

Основные возможности NumPy

✔ Многомерные массивы (`ndarray`) – более эффективные, чем списки Python.

✔ Векторизованные операции – быстрые вычисления без циклов.

✔ Линейная алгебра, статистика, рандомизация – встроенные математические функции.

✔ Интеграция с другими библиотеками (SciPy, pandas, TensorFlow).

Плюсы NumPy

🚀 Высокая производительность (оптимизированные низкоуровневые функции).

📊 Удобные операции с массивами (индексация, срезы, broadcasting).

🔢 Широкие возможности для математических вычислений.

2. pandas

pandas — библиотека для анализа и обработки табличных данных. Основные структуры:

- `Series` – одномерный массив с индексами (аналог столбца таблицы).

- `DataFrame` – двумерная таблица (как в Excel или SQL).

Основные возможности pandas

✔ Чтение/запись данных\*\* (CSV, Excel, SQL, JSON).

✔ Очистка данных (удаление пропусков, дубликатов).

✔ Агрегация и группировка (`groupby`, `pivot\_table`).

✔ Слияние и соединение данны (`merge`, `concat`).

✔ Временные ряды (работа с датами и временем).

Плюсы pandas

📋 Удобная работа с таблицами (фильтрация, сортировка, группировка).

🔍 Поддержка пропущенных значений (`NaN`).

⏱ Оптимизированные операции для больших данных.

3. Matplotlib

Matplotlib — библиотека для визуализации данных. Позволяет строить:

- Линейные графики (`plot`),

- Гистограммы (`hist`),

- Столбчатые диаграммы (`bar`),

- Круговые диаграммы (`pie`),

- 3D-графики и другие.

Основные возможности Matplotlib

✔ Гибкая настройка графиков (цвета, подписи, легенды).

✔ Поддержка разных стилей (`seaborn`, `ggplot`).

✔ Интерактивные графики (с `%matplotlib notebook` в Jupyter).

Плюсы Matplotlib

📈 Широкие возможности кастомизации.

🖼 Поддержка разных форматов (PNG, PDF, SVG).

🔌 Интеграция с `pandas` и `seaborn` для более красивых графиков.

Вывод

| Библиотека | Назначение | Ключевые особенности |

|------------------|---------------------------------------|----------------------------------------------------------|

| NumPy | Численные операции | Быстрые массивы, линейная алгебра |

| pandas | Анализ табличных данных | DataFrame, группировки, очистка данных|

| Matplotlib | Визуализация данных | Графики, гистограммы, кастомизация |