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

ФАКУЛЬТЕТ «Робототехники и комплексной автоматизации»
КАФЕДРА «Системы автоматизированного проектирования (РК-6)»

ОТЧЕТ О ВЫПОЛНЕНИИ ЛАБОРАТОРНОЙ РАБОТЫ

по дисциплине «Вычислительная математика»

Студент:	Лебедева Мария Дмитриевна
Группа:	РК6-52Б
Тип задания:	Лабораторная работа
Тема:	Модель биологического нейрона

Студент ________ подпись, дата Лебедева М.Д. Фамилия, И.О.

Преподаватель ________ подпись, дата ________ Фамилия, И.О.

Москва, 2021

Содержание

Модель биологического нейрона

1 Задание (Модель Ижикевича)

Дана система из двух ОДУ 1-го порядка:

$$\begin{cases} \frac{dv}{dt} = f_1(u,v) = 0.04v^2 + 5v + 140 - u + I, \\ \frac{du}{dt} = f_2(u,v) = a(bv - u); \end{cases} \tag{1}$$

и допалнительного условия, определяющего возникновения импульса в нейроне:

$$\text{если v} \geq 30\text{, то} \begin{cases} v \leftarrow c, \\ u \leftarrow u + d; \end{cases} \tag{2}$$

где v - потенциал потенциал мембраны (мВ), u– переменная восстановления мембраны (мВ),t– время (мс),I– внешний ток, приходящий через синапс в нейрон от всех нейронов,с которыми он связан.

Описания параметров представленной системы: a – задает временной масштаб для восстановления мембраны (чем больше a, темсбыстрее происходит восстановление после импульса); b– чувствительность переменной восстановления к флуктуациям разности по-тенциалов;

Таблица 1: Характерные режимы заданной динамической системы и соответствую-щие значения ее параметров

Режим	a	b	c	d
Tonic spiking (TS)	0.02	0.2	-65	6
Phasic spiking (PS)	0.02	0.25	-65	6
Chattering (C)	0.02	0.2	-50	2
Fast spiking (FS)	0.1	0.2	-65	2

c– значение потенциала мембраны сразу после импульса; d– значение переменной восстановления мембраны сразу после импульса.

Требуется (базовая часть):

1. Реализовать следующие функции, каждая из которых возвращает дискретнуютраекторию системы ОДУ с правой частью, заданной функцией **f**, начальным условием **x_0**, шагом по времени **h** и конечным временем **t_n**:

–**euler(x_0, t_n, f, h)**, где дискретная траектория строится с помощью методаЭйлера;

–**implicit_euler(x_0, t_n, f, h)**, где дискретная траектория строится с помощьюнеявного метода Эйлера;

–**runge_kutta(x_0, t_n, f, h)**, где дискретная траектория строится с помощьюметода Рунге–Кутта 4-го порядка.

2. Для каждого из реализованных методов численно найти траектории заданнойдинамической системы, используя шаг $h = 0.5$ и характерные режимы, указанные в таблице 1. В качестве начальных условий можно использовать $v(0) = c$и $u(0){=}bv(0)$.Внешний ток принимается равным $I{=}\ 5$.

3. Вывести полученные траектории на четырех отдельных графиках как зависимости потенциала мембраны v от времени t, где каждый график должен соответствовать своему характерному режиму работы нейрона.

4. По полученным графикам кратко описать особенности указанных режимов.

2 Цель выполнения лабораторной работы

В базовой части данной работы предлагается познакомиться с мотодами решения задачи Коши для систем обыкновенных дифференциальных уравнений (ОДУ) 1-го порядка например модели Ижекевича.

3 Выполненные задачи

1. На языке программирования Python были написаны функции, вычисляющие:

– дискретную траекторию, которая строится с помощью метода Эйлера;

– дискретную траекторию, которая строится с помощью неявного метода Эйлера;

– дискретную траекторию, которая строится с помощьюметода Рунге–Кутта 4-го порядка.

2. Для каждого из реализованных методов численно найдена траектория заданной динамической системы.

3. Построены графики зависимости потенциала мембраны v от времени t.

4. Проведен анализ по полученным графикам для описания особенностей указанных режимов.

4 Разработка функций для возвращения дискретной траектории системы ОДУ

Для решения системы дифференциальных уравнений воспользуемся методом Эйлера, неявным методом Эйлера и методом Рунге-Кутта 4-го порядка.
Для решения методом Эйлера воспользуемся формулой:

$$w_{i+1} = w_i + hT(t_i, w_i), \tag{3}$$

где $w_i \approx y(t_i)$, h - шаг, T - система дифференциальных уравнений (1), $t_i = a + ih$, $i = 0, 1, ..., m$. Учитывая, что заданы два дифференциальных уравнений, которые являются системой ОДУ, представим w, в виде вектора $w_i = [y_1(t_i), y_2(t_i)]$.

Листинг 1. Алгоритм метода Эйлера.

```
def euler(u_0, v_0, t_n, h, I, mode):
    n = int(round(t_n / h))    # Количество узлов
    u = np.zeros((n + 1))    # Массив для решений
    v = np.zeros((n + 1))  # Массив для решений
    t = np.linspace(0, n * h, n+1)    # Генерация дискретных значений по оси времени
    u[0] = u_0
    v[0] = v_0
    for i in range(0,n):
        if v[i] >= 30:
            v[i] = modes[mode][2]
            u[i] += modes[mode][3]
        v_f, u_f = f(v[i], u[i], I, mode)
        v[i + 1] = v[i] + h * v_f
```

```
        u[i + 1] = u[i] + h * u_f
    return t, v, u
```

Для решения неявным методом Эйлера воспользуемся формулой:

$$w_{i+1} = w_i + hT(t_i, w_{i+1}), \tag{4}$$

где $w_i \approx y(t_i)$, h - шаг, T - система дифференциальных уравнений (1), $t_i = a + ih$, $i = 0, 1, ..., m$.

Листинг 2. Алгоритм неявного метода Эйлера.

```
def imp_euler(u_0, v_0, t_n, h, I, mode):
    n = int(round(t_n / h))   # Количество узлов
    u = np.zeros((n + 1))   # Массив для решений
    v = np.zeros((n + 1))  # Массив для решений
    t = np.linspace(0, n * h, n+1)   # Генерация дискретных значений по оси времен
    u[0] = u_0
    v[0] = v_0

    def phiV(vi_1, ui, vi):
        return vi_1 - vi - h * f(v[i], u[i], I, mode)[0]

    def phiU(ui_1, ui, vi):
        return ui_1 - ui - h * f(v[i], u[i], I, mode)[1]

    for i in range(n):
        v[i+1] = optimize.fsolve(phiV, v[i], args=(u[i], v[i]))
        u[i+1] = optimize.fsolve(phiU, u[i], args=(u[i], v[i]))
        if v[i+1] >= 30:
            v[i+1] = modes[mode][2]
            u[i+1] = u[i+1] + modes[mode][3]
    return t, v, u
```

Запишем метод Рунге-Кутта 4-го порядка:

$$w_0 = \alpha \tag{5}$$

$$k_1 = hT(t_i, w_i) \tag{6}$$

$$k_2 = hT(t_i + \frac{h}{2}, k_1 + \frac{h}{2}) \tag{7}$$

$$k_3 = hT(t_i + \frac{h}{2}, k_2 i + \frac{h}{2}) \tag{8}$$

$$k_1 = hT(t_i + \frac{h}{2}, w_i + k_3) \tag{9}$$

$$\frac{w_{i+1} = w_i + 1}{6(k_1 + 2k_2 + 2k_3 + k_4), i = 0, 1, ..., m - 1} \tag{10}$$

Листинг 3. Алгоритм метода Рунге-Кутта 4-го порядка.

```python
def runge_kutta(u_0, v_0, t_n, h, I, mode):
    n = int(round(t_n / h))  # Количество узлов
    u = np.zeros((n + 1))  # Массив для решений
    v = np.zeros((n + 1))  # Массив для решений
    t = np.linspace(0, n * h, n + 1)
    u[0] = u_0
    v[0] = v_0

    for i in range(n):
        k1_v = h * f(v[i], u[i], I, mode)[0]
        k1_u = h * f(v[i], u[i], I, mode)[1]

        k2_v = h * f(v[i] + k1_v / 2, u[i] + h / 2, I, mode)[0]
        k2_u = h * f(v[i] + h / 2, u[i] + k1_u / 2, I, mode)[1]

        k3_v = h * f(v[i] + k2_v / 2, u[i] + h / 2, I, mode)[0]
        k3_u = h * f(v[i] + h / 2, u[i] + k2_u / 2, I, mode)[1]

        k4_v = h * f(v[i] + k3_v, u[i] + h, I, mode)[0]
        k4_u = h * f(v[i] + h, u[i] + k3_u, I, mode)[1]

        v[i + 1] = v[i] + (k1_v + 2 * k2_v + 2 * k3_v + k4_v) / 6
        u[i + 1] = u[i] + (k1_u + 2 * k2_u + 2 * k3_u + k4_u) / 6
        if v[i + 1] >= 30:
            v[i + 1] = modes[mode][2]
            u[i + 1] += modes[mode][3]

    return t, v, u
```

5 Построение графиков зависимости потенциала v от премени t

Возьмем промежуток времени $t \in [0; 300]$, шаг $h = 0.5$ и параметры a, b, c, d, I, которые нам даны по условию. Каждый график соответствует одгому из режимов в таблице 1

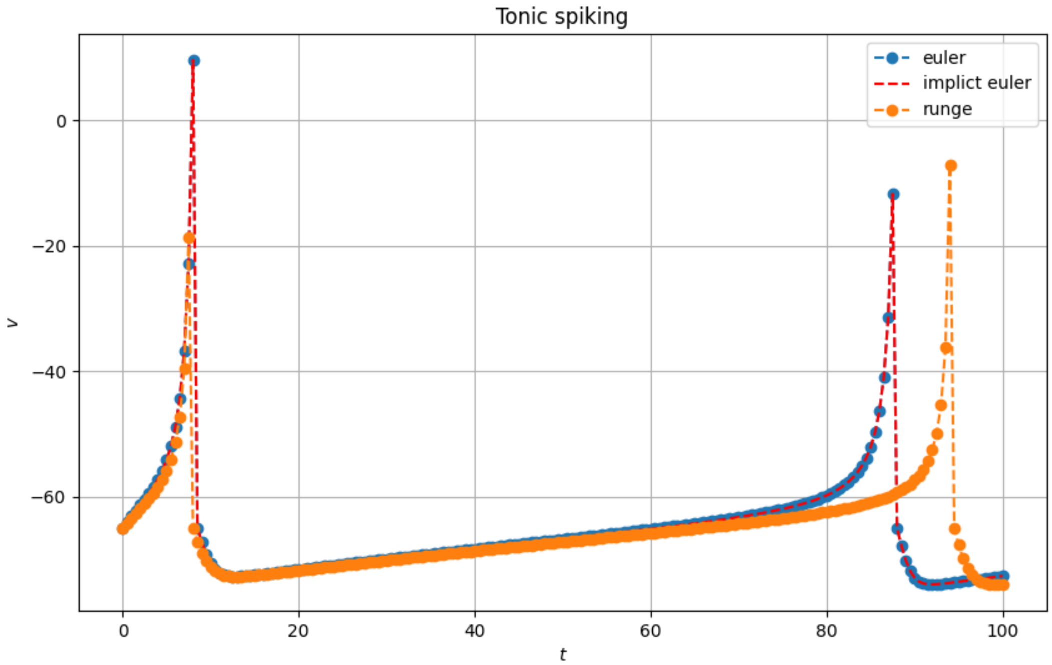

Рисунок 1. Траектории для метода "Tonic Spiking".

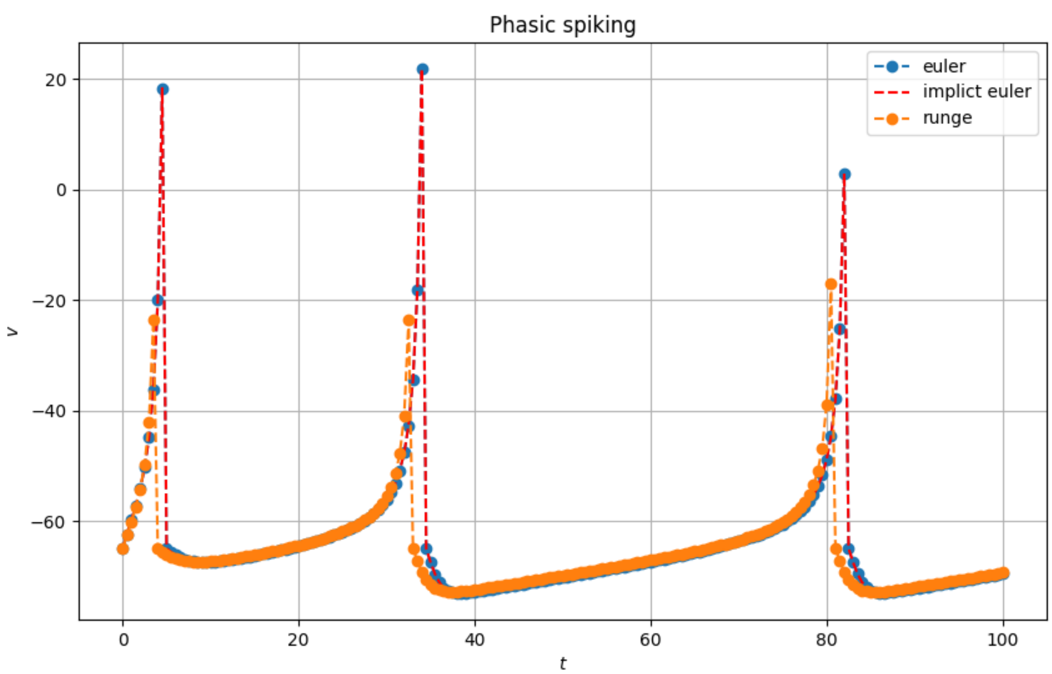

Рисунок 2. Траектории для метода "Phasic Spiking".

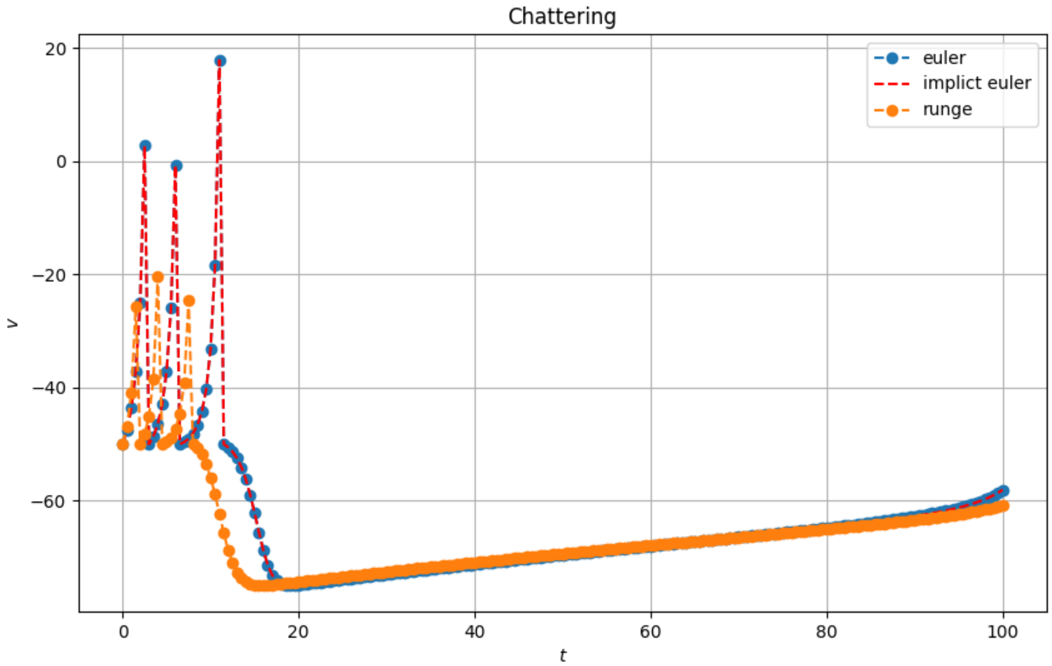

Рисунок 3. Траектории для метода "Chatter".

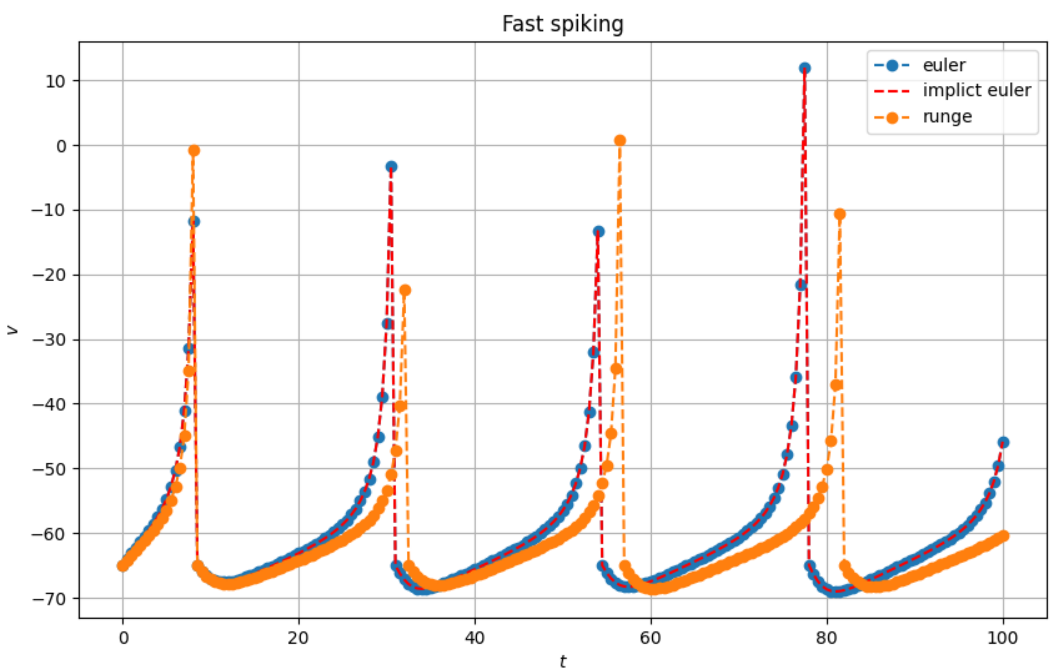

Рисунок 4. Траектории для метода "Fast Spiking".

Листинг 4. Построение графиков

```
def composite_trapezoid(f, a, b, n):
    if n % 2 != 0:
        n += 1
        x = np.linspace(a, b, n + 1)
        h = (b - a) / n
        return h/2. * (f(x[0]) + 2*np.sum(f(x[1:-1:1]))+ f(x[-1]))
```

6 Анализ описания особенностей указанных режимов

В режиме "Tonic Spiking рисунок 1, низкая частота пульсаций, что обуславливается большим временем восстановленя мембранны после импульса и её маленьким временным масштабом сосстановления.

В режиме "Phasic Spiking рисунок 2, частота пульсаций больше, чем в предыдущем режиме. Это обуславливается тем, что значение переменной восстановления к фуктуациям разности потенциалов выше.

В режиме "Chattering рисунок 3, частота пульсаций похожа на частоту пульсаций в режиме "Tonic Spiking"(Рис. 1). Но в момент импульса график не схож ни с одним из других режимов. Это происходит из-за большего значения потенциала после импульса.

В режиме "Fast Spiking рисунок 4, частота пульсаций еще больше, так как время восстановления мембраны после импульса больше.

7 Заключение

В лабораторной работе были написаны функции написаны функции, вычисляющие дискретные траектории, вычисляющиеся с помощью мотода Эйлера, неявного метода Эйлера и метода Рунге-Кутта 4-го порядка. Найдена траектория заданной динамисеской системы для каждого из методов. Построены графики зависимости потенциала мембраны v от времени t.

А так же определен оптимальный метод для решения задачи Коши. Которым является метод Рунге-Катта 4-го порядка, потому что имеет малый порядок погрешности.

Список использованных источников

1. Першин А.Ю. Лекции по курсу «Вычислительная математика». Москва, 2018-2021. С. 140.

Выходные данные

Лебедева М.Д.. Отчет о выполнении лабораторной работы по дисциплине «Вычислительная математика». [Электронный ресурс] — Москва: 2021. — 11 с. URL: https://sa2systems.ru:88 (система контроля версий кафедры РК6)

Постановка: © *ассистент кафедры РК-6, PhD А.Ю. Першин*

Решение и вёрстка: © *студент группы РК6-52Б, Лебедева М.Д.*

2021, осенний семестр