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:	Караф Сармат Майк
Группа:	РК6-52Б
Тип задания:	лабораторная работа
Тема:	Модель биологического нейрона

Студент _____ (подпись, дата) Караф С. М. (Фамилия, И.О.)

Преподаватель _____ (подпись, дата) Соколов А. П. (Фамилия, И.О.)

Москва, 2021

Содержание

Модель биологического нейрона

1 Задание

Численные методы решения задачи Коши для систем обыкновенных дифференциальных уравнений (ОДУ) 1-го порядка активно используются далеко за пределами стандартных инженерных задач. Примером области, где подобные численные методы крайне востребованы, является нейробиология, где открытые в XX веке модели биологических нейронов выражаются через дифференциальные уравнения 1-го порядка. Математическая формализация моделей биологических нейронов также привела к появлению наиболее реалистичных архитектур нейронных сетей, известных как спайковые нейронные сети (Spiking Neural Networks). В данной лабораторной работе мы исследуем одну из простейших моделей подобного типа: 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 дополнительного условия, определяющего возникновение 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 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, тем быстрее происходит восстановление после импульса);

b – чувствительность переменной восстановления к флуктуациям разности потенциалов;

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 дискретную траекторию 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 метода Эйлера;

   -$implicit\_euler(x\_0, t\_n, f, h)$, где дискретная траектория строится с помощью неявного метода Эйлера;

   -$runge\_kutta(x\_0, t\_n, f, h)$, где дискретная траектория строится с помощью метода Рунге–Кутта 4-го порядка.

2. Для каждого из реализованных методов численно найти траектории заданной динамической 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.

Требуется (продвинутая часть).

1. Объяснить, в чем состоят принципиальные отличия реализованных методов? В чем они схожи?

2. Произвести интегрирование во времени до 1000 мс нейронной сети с помощью метода Эйлера, используя следующую информацию.

   (a) Динамика каждого нейрона в нейронной сети описывается заданной моделью Ижикевича. В нейронной сети имеется 800 возбуждающих нейронов и 200 тормозных. Возбуждающие нейроны имеют следующие значения параметров: $a = 0.02$, $b = 0.2$, $c = -65 + 15\alpha^2$, $d = 8 - 6\beta^2$ и внешний ток в отсутствие токов от других нейронов равен $I = I_0 = 5\xi$, где α, β и ξ – случайные числа от 0 до 1 (распределение равномерное). Тормозные нейроны имеют следующие значения параметров: $a = 0.02 + 0.08\gamma$, $b = 0.25 - 0.5\delta$, $c = -65$, $d = 2$ и внешний ток в отсутствие токов от других нейронов равен $I = I_0 = 2\zeta$, где γ, δ и ζ – случайные числа от 0 до 1. В качестве начальных условий используются значения $v(0) = -65$ и $u(0) = bv(0)$.

(b) Нейронная сети может быть смоделирована с помощью полного графа. Матрица смежности W этого графа описывает значения токов, передаваемых от нейрона к нейрону в случае возникновения импульса. То есть, при возникновении импульса нейрона j внешний ток связанного с ним нейрона i единовременно увеличивается на величину W_{ij} и затем сразу же падает до нуля, что и моделирует передачу импульса по нейронной сети. Значение W_{ij} равно 0.5θ, если нейрон j является возбуждающим, и $-\tau$, если тормозным, где θ и τ – случайные числа от 0 до 1.

3. Вывести на экран импульсы всех нейронов как функцию времени и определить частоты характерных синхронных (или частично синхронных) колебаний нейронов в сети.

2 Цель выполнения лабораторной работы

Цель выполнения лабораторной работы – исследование работы модели биологического нейрона, на примере модели Ижикевича, в которой происходят процессы генерации и распространения нейронных импульсов.

3 Выполненные задачи

Базовая часть:

1. Разработаны функции, которые возвращают дискретную траекторию системы ОДУ, при помощи метода Эйлера, неявного метода Эйлера, метода Рунге–Кутта 4-го порядка.

2. Численное нахождение траектории заданной динамической системы. Вывод полученных траекторий на графиках, как зависимости потенциала мембраны v от времени t, где каждый график должен соответствовать своему характерному режиму работы нейрона.

3. Описание особенностей указанных режимов из таблицы.

Продвинутая часть:

1. Анализ реализованных методов

2. Произведены интегрирование во времени до 1000мс нейронной сети с помощью метода Эйлера для 800 возбуждающих нейронов и 200 тормозных с заданными параметрами. Сгенерирована матрица смежности, имитирующая передачу импульсов между нейронами.

3. Построен график импульсов всех нейронов в течении 1000мс, и определены частоты частично синхронных колебаний нейронов в сети.

4 Базовая часть

4.1 Разработка функций, метода Эйлера, неявного Эйлера и Рунге–Кутта, для постоении траекторию системы ОДУ.

Исследуя модель биологического нейрона, были реализованные функции, которые возвращает дискретную траекторию системы ОДУ .
Рассмотрим задачу Коши для системы обыкновенных дифференциальных уравнений:

$$\begin{cases} \frac{dv}{dt} = f_1(u,v) \\ \frac{du}{dt} = f_2(u,v) \end{cases},$$

где $t \in [0;T]$, T – конечное время, f_1, f_2 – заданная функция $v(0) = c$ и $u(0) = bv(0)$, рассматриваемые методы, предполагают дискретизацию координаты t вида $t_i = ih$, $i = 1,\ldots,m$, где $h = \frac{T}{m}$ называют шагом.

Разработаем функцию, которая возвращает решение системы ОДУ (1), где mode - режим работы нейрона, реализация функции *def f(u_f, v_f, mode)* показаны в файле Base_lab3.py и в листинге 1.

Листинг 1. Функция решения системы ОДУ (1)

```
1 def f(u_f, v_f, mode):
2     v = 0.04 * v_f ** 2 + 5 * v_f + 140 - u_f + I
3     u = coef[mode][0] * (coef[mode][1] * v_f - u_f)
4     return np.array([v, u])
```

Реализация функции метода Эйлера *def euler(x_0, t_n, f, h, mode)*, для решения задачи Коши, основана на модели Ижикевича.

Формулировка метод Эйлера имеет следующий вид:

$$\begin{cases} v_{i+1} = v_i + hf_1(u_i, v_i) \\ u_{i+1} = u_i + hf_2(u_i, v_i), \ i = 0,1,...,m-1 \end{cases} \tag{3}$$

где v - потенциал мембраны, u - переменная восстановление мембраны, $t \in [0;T]$, T – конечное время, $f_{1,2}$– заданная функция $v(0) = c$ и $u(0) = bv(0)$, $t_i = ih, i = 1,\ldots,m$, $h = \frac{T}{m}$.

Программная реализация функция метода Эйлера *def euler(x_0, t_n, f, h, mode)*, представлена в файле Base_lab3.py и в листинге 2, представляет собой цикл, в котором на каждом интерации высчитывается значение v_i и u_i по времени интегрировании t, по завершению работы функция возвращает значение потенциала мембраны v и время интегрирования .

Листинг 2. Функция метода Эйлера для определения траектории системы ОДУ (1)

```
def euler(x_0, t_n, f, h, mode):
    v = [x_0[0]]
    u = [x_0[1]]
    m = int(t_n / h)
    t = np.linspace(0, t_n, m + 1)
    for i in range(0, m):
        if v[i] >= 30:
            v[i] = coef[mode][2]
            u[i] += coef[mode][3]
        v_f, u_f = f(u[i], v[i], mode)
        v.append(v[i] + h * v_f)
        u.append(u[i] + h * u_f)
    return t, v
```

Реализация Функции неявного метода Эйлера *implicit_ euler(x_ 0, t_ n, f, h, mode)*, для решения задачи Коши, основана на модели Ижикевича.

Неявный метод Эйлера можно представить, как p-шаговый метод Адамса-Моултона (4), при $p = 2$:

$$w_0 = \tilde{\alpha}_0, \quad w_1 = \tilde{\alpha}_1, \ \ldots, \quad w_{p-2} = \tilde{\alpha}_{p-2},$$
$$w_{i+1} = w_i + h \sum_{j=1}^{p} f(t_{i-j+2}, w_{i_{i-j+2}}), \quad i = p-2, p-1, ..., m-1, \tag{4}$$

где $\tilde{\alpha}$ - начально значение, $t_i \in [a; b]$, $t_i = a + ih = 1, \ldots, m$, $h = \frac{b-a}{m}$

От сюда неявных метод Эйлера (5) в многомерном случае примет вид:

$$w_0 = \tilde{\alpha}_0$$
$$w_{i+1} = w_i + hf(t_{i+1}, w_{i_{i+1}}), \quad i = 0, 1, ..., m-1 \tag{5}$$

где $\tilde{\alpha}$ - начально значение, $t_i \in [a; b]$, $t_i = a + ih = 1, \ldots, m$, $h = \frac{b-a}{m}$

В рамках данной лабораторной работы, неявный метод Эйлера примет вид:

$$\begin{cases} v_{i+1} = v_i + hf_1(u_i, v_{i+1}) \\ u_{i+1} = u_i + hf_2(u_{i+1}, v_i), \quad i = 0, 1, ..., m-1 \end{cases} \tag{6}$$

где v - потенциал мембраны, u - переменная восстановление мембраны, $t \in [0; T]$, T – конечное время, $f_{1,2}$– заданная функция $v(0) = c$ и $u(0) = bv(0)$, $t_i = ih, i = 1, \ldots, m$, $h = \frac{T}{m}$.

Программная реализация функция неявного метода Эйлера *implicit_ euler(x_ 0, t_ n, f, h, mode)*, представлена в файле *Base_ lab3.py* и в листинге 3, представляет собой решение нелинейных алгебраических уравнений, для решения которых воспользуемся вспомогательными средствами, функцией *root* из библиотеки *scipy optimize*. Реализация данной функции схож с реализацией функции явного Эйлера, на каждой итерации высчитывается значение v_i и u_i по времени интегрировании t.

Листинг 3. Функция неявного метода Эйлера для определения траектории системы ОДУ (1)

```
def implicit_euler(x_0, t_n, f, h, mode):
    m = int(t_n / h)
    v = np.zeros((m + 1))
    u = np.zeros((m + 1))
    v[0] = x_0[0]
    u[0] = x_0[1]
    for k in range(0, m):
        slov = optimize.root(pvu, (v[k], u[k]), args=(u[k], v[k]))
        v[k + 1] = slov.x[0]
        u[k + 1] = slov.x[1]
        if v[k + 1] >= 30:
            v[k + 1] = coef[mode][2]
            u[k + 1] += coef[mode][3]
    return t_n, v
```

Было реализована функция *pvu(vu_1, ui, vi)* листинг 4, для решения нелинейных уравнений. По завершению работы *implicit_euler(x_0, t_n, f, h, mode)* функция возвращает значение потенциала мембраны и время интегрирования.

Листинг 4. Функция решения нелинейных уравнений для *scipy optimize*

```
    def pvu(vu_1, ui, vi):
        v_s = vu_1[0] - vi - h * f(ui, vu_1[0], mode)[0]
        u_s = vu_1[1] - ui - h * f(vu_1[1], vi, mode)[1]
        return [v_s, u_s]
```

Реализация Функции метода Рунге–Кутта 4-го порядка *runge_kutta(x_0, t_n, f, h, mode)*, для решения задачи Коши, основана на модели Ижикевича.

Формулировка метод Рунге–Кутта 4-го порядка имеет следующий вид:

$$\begin{cases} k_{1v} = hf_1(u_i, v_i), \\ k_{1u} = hf_2(u_i, v_i), \\ k_{2v} = hf_1(u_i + \frac{k_{1u}}{2}, v_i + \frac{k_{1v}}{2}), \\ k_{2u} = hf_2(u_i + \frac{k_{1u}}{2}, v_i + \frac{k_{1v}}{2}), \\ k_{3v} = hf_1(u_i + \frac{k_{2u}}{2}, v_i + \frac{k_{2v}}{2}), \\ k_{3u} = hf_2(u_i + \frac{k_{2u}}{2}, v_i + \frac{k_{2v}}{2}), \\ k_{4v} = hf_1(u_i + k_{3u}, v_i + k_{3v}), \\ k_{4u} = hf_2(u_i + k_{3u}, v_i + k_{3v}), \\ v_{i+1} = v_i + \frac{1}{6}(k_{1v} + 2k_{2v} + 2k_{3v} + k_{4v}), \\ u_{i+1} = u_i + \frac{1}{6}(k_{1u} + 2k_{2u} + 2k_{3u} + k_{4u}), \ i = 0, 1, ..., m-1. \end{cases} \tag{7}$$

где v - потенциал мембраны, u - переменная восстановление мембраны, $t \in [0; T]$, T – конечное время, $f_{1,2}$– заданная функция $v(0) = c$ и $u(0) = bv(0)$, $t_i = ih, i = 1, \ldots, m$, $h = \frac{T}{m}$.

Программная реализация функция метода Рунге–Кутта 4-го порядка *runge_kutta(x_0, t_n, f, h, mode):*, представлена в файле Base_lab3.py и в листинге 5, представляет собой цикл, в котором вычисляться значение k_1, k_2, k_3, k_4 на каждой итерации, и при помощи них высчитываются значение v_i и u_i по времени интегрировании t. По завершению работы функция возвращает значение потенциала мембраны и время интегрирования.

Листинг 5. Функция метода Рунге–Кутта 4-го порядка для определения траектории системы ОДУ (1)

```
def runge(x_0, t_n, f, h, mode):
    v = [x_0[0]]
    u = [x_0[1]]
    m = int(t_n / h)
    t = np.linspace(0, t_n, m + 1)
    for j in range(0, m):
        if v[j] >= 30:
            v[j] = coef[mode][2]
            u[j] += coef[mode][3]
        k1 = h * f(u[j], v[j], mode)
        k2 = h * f(u[j] + 0.5 * k1[1], v[j] + 0.5 * k1[0], mode)
        k3 = h * f(u[j] + 0.5 * k2[1], v[j] + 0.5 * k2[0], mode)
        k4 = h * f(u[j] + k3[1], v[j] + k3[0], mode)
        u.append(u[j] + (1 / 6) * (k1[1] + 2 * k2[1] + 2 * k3[1] + k4[1]))
        v.append(v[j] + (1 / 6) * (k1[0] + 2 * k2[0] + 2 * k3[0] + k4[0]))
    return t, v
```

4.2 Нахождение траектории заданной динамической системы и построение графиков траектории для каждого режима нейрона.

Для построение графиков траектории заданной динамической системы воспользуемся функциями разработанные в предыдущем пункте, параметриту их в зависимости от режимов работы нейрона из таблицы 1 , с шагом $h = 0.1$, конечным временем $T = 250$.

Алгоритм нахождения траектории представляет собой цикл, в котором каждая итерация является режимом работы нейрона, передовая в функции методов решение ОДУ соответствующие коэффициенты. Данный алгоритм представлен в листинге 6 и в файле *Base_lab3.py*

Листинг 6. Алгоритм построение графиков траектории заданной динамической системы

```
name = ['Tonic spiking', 'Phasic spiking', 'Chattering', 'Fast spiking']
for i in range(4):
    plt.subplots(figsize=(13, 8))
    x_0 = [coef[i][2], coef[i][2] * coef[i][1]]
    t_e, y_e = euler(x_0, t_n, f, h, i)
    t_r, y_r = runge(x_0, t_n, f, h, i)
    t_be, y_be = backward_euler(x_0, t_n, f, h, i)
    plt.title(name[i], loc='left')
    plt.plot(t_e, y_e, 'o--', label=r'euler')
    plt.plot(t_be, y_be, 'o--', label=r'back euler')
    plt.plot(t_r, y_r, 'o--', label=r'runge')
    plt.xlabel(r'$t$', fontsize=16)
    plt.ylabel(r'$y$', fontsize=16)
    plt.legend()
    plt.grid()
    plt.show()
```

По окончанию алгоритма получаем графики (Рис. 1, 2, 3, 4) для характерного режима работы нейрона, с зависимостью потенциала мембраны v от времени t.

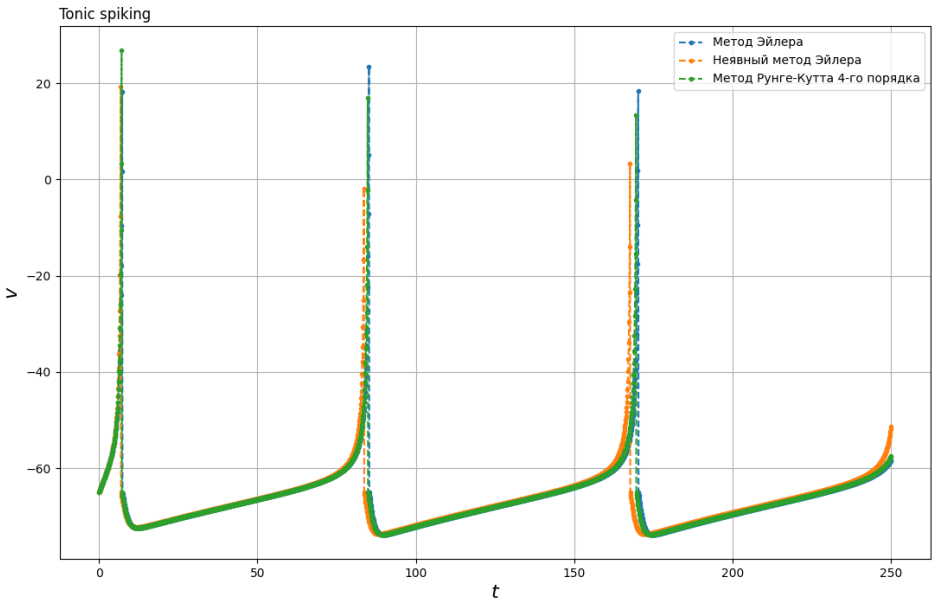

Рис. 1. Траектория работы нейрона в режиме Tonic spiking.

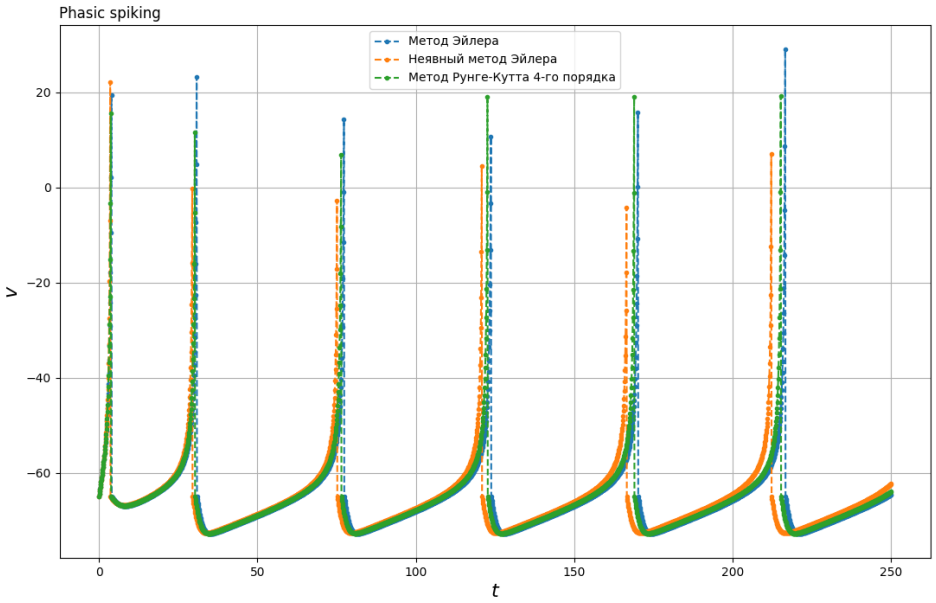

Рис. 2. Траектория работы нейрона в режиме Phasic spiking.

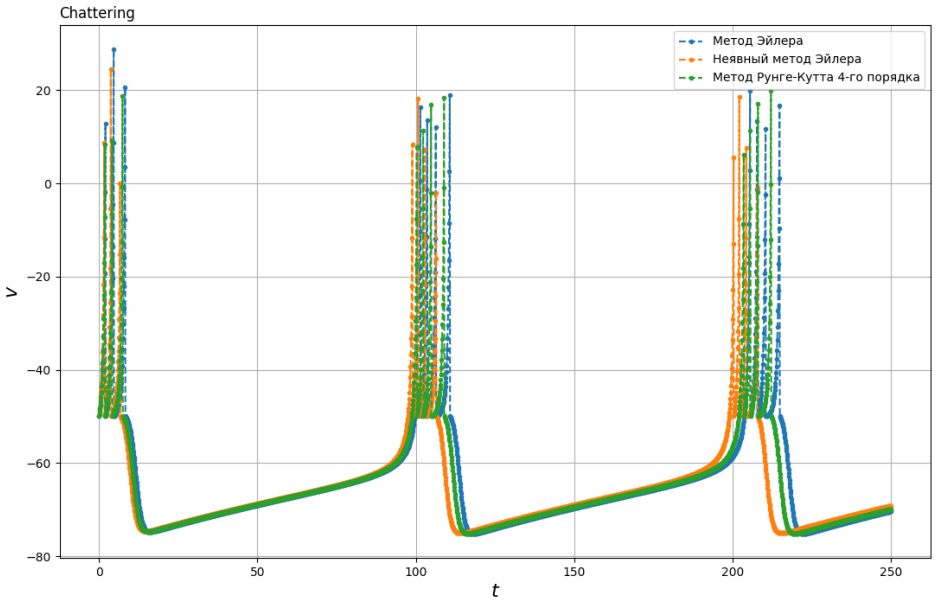

Рис. 3. Траектория работы нейрона в режиме Chattering.

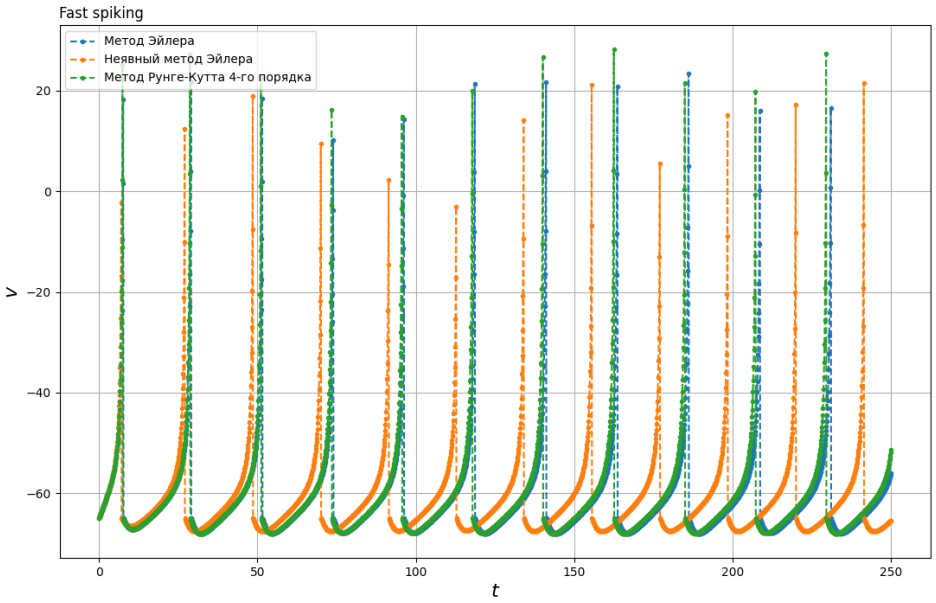

Рис. 4. Траектория работы нейрона в режиме Fast spiking.

4.3 Описание особенностей режимов работы нейрон.

По ранее полученным графикам (Рис. 1, 2, 3, 4), можно заметить особенности режимов работы. Импульсы показанные на графиков показывают скорость восстановление мембраны, от сюда можно сделать вывод, что у режима *Fast spiking* (Рис.4) самое быстрое восстановление мембраны, а у режима *Chattering* (Рис.3) самое медленное восстановление мембраны. Из всех режимов выделяется Chattering (Рис 3), восстановление мембраны в нём происходят с определёнными сериями, и сами серии происходят с определённой периодичностью, так же в данном режиме самый высокий потенциал мембраны. Из графика (Рис.1, 2) режимов *Tonic spiking* и *Phasic spiking*, заметно что данные режимы имеют средние значение восстановительной мембраны и потенциал мембраны.

5 Продвинутая часть

5.1 Анализ реализованных методов.

Из курса вычислительной математики знаем, что наивысшую точность имеет метод Рунге–Кутта 4-го порядка. Для проведения анализов решения задачи Коши для систем ОДУ, возьмем метод Рунге–Кутта 4-го порядка за аналитическое решение, из-за сложности вычисления аналитического решения ОДУ. Глобальные погрешности для метода Эйлера $O(h)$, неявный метод Эйлера $O(h^2)$, метод Рунге–Кутта $O(h^4)$. На графиках

(Рис. 1, 2, 3, 4) , можно заметить что с течением времени глобальная погрешность методов накапливается, тесть с окончанием времени глобальная погрешность у методов максимальная. Хоть и у неявного метода Эйлера точность больше, чем у явного метода Эйлера, но на графиках можно наблюдать, что отклонение от Рунге-Кутта, больше чем у Эйлера. Это обусловлено вычислительной сложностью, что на каждой итерации требуется решить в общем случае нелинейное уравнение. В данной лабораторной работе метод Эйлер является самым оптимальным, хотя и обладает большой глобальной погрешностью.

5.2 Моделирование нейронной сети с помощью метода Эйлера для 800 возбуждающих нейронов и 200 тормозных.

Для моделирования неровной сети, нужно сгенерировать 1000 нейронов, где 800 возбуждающих и 200 тормозных с определение параметрами, так же необходимо задать матрицу смежности W_{ij} (8), которая описывает значение токов, передаваемых от нейрона к нейрону. Матрица смежности заполняется по следующим правилам если значение W_{ij} равно0.5θ, если нейрон j является возбуждающими $-\tau$, если тормозным, где θ, τ - случайные величины

$$W_{ij} = \begin{vmatrix} 0 & 0.5\theta & \dots & -\tau & -\tau \\ 0.5\theta & 0 & \dots & -\tau & -\tau \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0.5\theta & 0.5\theta & \dots & 0 & -\tau \\ 0.5\theta & 0.5\theta & \dots & -\tau & 0 \end{vmatrix}, \tag{8}$$

Программная реализация функции *time_impulse(Voz, Torm, t_n, x_0)*, представлена в файле Prof_lab3.py и в листинге 7, представляет собой цикл интегрирование по времени t с шагом равным $h = 0.5$, в котором задаться массив отображающий наличие импульса в каждой итерации. При возникновении импульса, и выполнении условии 2, переопределяться значение v, u. Далее полученное время и индекс нейронов записываться в массив, где первым элементом является время возникновения импульса, а второй индекс нейронов у которых возник импульс в это время. После записи времени импульса суммируются токи значение из матрицы смежности с первоначальным током нейрона. Используя метод Эйлера (3) для системы ОДУ (1) будет вычисляться дискретная траектория. По завершению работы функция возвращает массив t_imp зависимость времени импульса от индекса нейрона.

Листинг 7. Функция моделирования нероной сети по модели Иэикевича

```
def time_impulse(Voz, Torm, t_n, x_0):
    t_imp = []
    v, u = x_0
    kc_i = np.random.uniform(0, 1, Voz)
    k0_i = np.random.uniform(0, 1, Torm)
    for t in np.arange(t_n, step=0.5):
        imp = v > 30
        v[imp] = coef[2][imp]
        u[imp] = u[imp] + coef[3][imp]
        n_imp = np.where(imp)
        if len(n_imp[0]) > 0:
            t_imp.append([t, n_imp[0]])
        I_0 = np.hstack((5 * kc_i, 2 * k0_i))
        I_0 += np.sum(W[:, imp], axis=1)
        v, u = f(v, u, I_0)
    return t_imp
```

5.2 Построение графиков импульсов всех нейронов, и определение характерных синхронных колебаний нейронов в сети.

Из полученных данных предыдущего пункта можно построить зависимость времени возникновения импульса от индекса нейрона (Рис. 5).

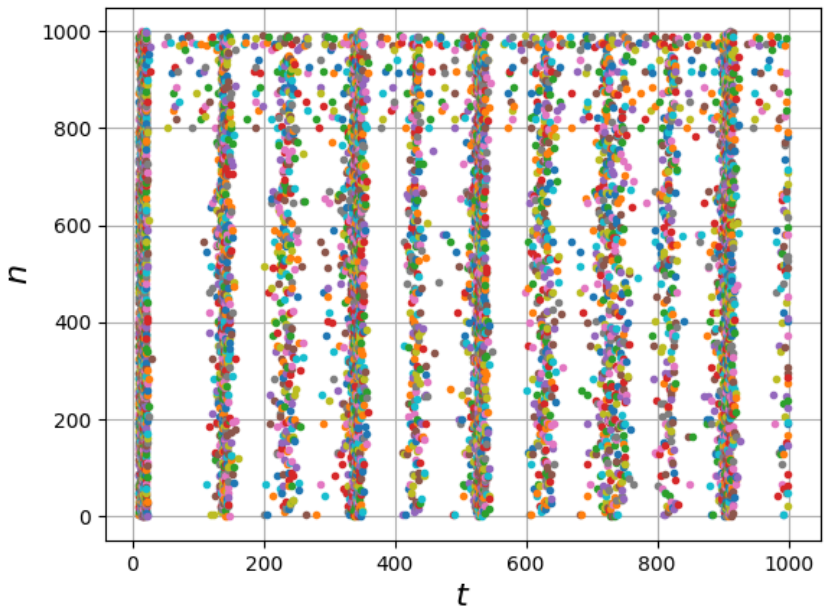

Рис. 5. Зависимость времени возникновения импульса от индекса нейрона.

Из курса физики известно, что частота колебаний - количественная характеристика периодических колебаний, равная отношению числа циклов колебаний ко времени их совершения, частота - величина, обратная периоду колебаний. Из графика (Рис. 5), приблизительный период равен 100 мс, от сюда следует что частота синхронных колебаний приблизительно равна 10 Гц.

6 Заключение

1. Реализованы функции методов численного решения задачи Коши для ОДУ 1-го порядка.

2. Построены графики дискретных траекторий методов для каждого режима отражающие работу нейронов: *Tonic spiking*, *Phasic spiking*, *Chattering*,*Fast spiking*. Были описаны особенности режимов работы нейрон

3. Было выявлено, что оптимальным решениям является метод Рунге-Кутта 4-го порядка

4. В результате моделирования нейронной сети по модели Ижикевича было выявлено колебания частотой 10 Гц.

Список использованных источников

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

2. Документация Python 3.9 [Электронный ресурс] [Офиц. сайт]. 2021. (дата обращения 27.10.2021).

Выходные данные

Караф С. М.. Отчет о выполнении лабораторной работы по дисциплине «Вычислительная математика». [Электронный ресурс] — Москва: 2021. — 15 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, Караф С. М.*

2021, осенний семестр