



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:	Дунайцев Александр Иванович
Группа:	РК6-54Б
Тип задания:	лабораторная работа
Тема:	Модель биологического нейрона

Студент

\_\_\_\_\_  
подпись, дата

Дунайцев А. И.  
\_\_\_\_\_  
Фамилия, И.О.

Преподаватель

\_\_\_\_\_  
подпись, дата

\_\_\_\_\_  
Фамилия, И.О.

Москва, 2021

# Содержание

<b>Модель биологического нейрона</b>	<b>3</b>
1    Задание . . . . .	3
2    Цель выполнения лабораторной работы . . . . .	5
3    Разработка функций. . . . .	5
Реализация функции <code>euler</code> . . . . .	6
Реализация функции <code>implicit_euler</code> . . . . .	6
Реализация функции <code>runge_kutta</code> . . . . .	7
4    Расчет траекторий динамической системы при использовании разных характерных режимов. Построение графиков полученных траекторий. .	8
5    Анализ построенных графиков. . . . .	11
6    Анализ методов решения задачи Коши для систем ОДУ. . . . .	11
7    Моделирование работы нейронной сети. . . . .	12
<b>1   Заключение</b>	<b>13</b>

# Модель биологического нейрона

## 1 Задание

В лабораторной работе исследуется модель биологического нейрона, а именно модель Ижикевича. Дана система из двух ОДУ 1-го порядка:

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

и дополнительного условия, определяющего возникновение импульса в нейроне:

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

где  $v$  потенциал мембраны (мВ),  $u$  - переменная восстановления мембраны (мВ),  $t$  - время (мс), внешний ток, проходящий 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;

$c$  - 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 ОДУ с правой частью, заданной функцией  $f$ , начальным условием  $x_0$ , шагом по времени  $h$  и 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-Кутты 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  $I = 5$ .

Таблица 1. Характерные режимы заданной динамической системы и соответствующие значения ее параметров.

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

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. Для каждой из схем каково значение шага, при котором она становится неустойчивой?
2. Произвести интегрирование по времени до 1000 мс нейронной сети с помощью метода Эйлера, используя следующую информацию.
  - 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 других токов от других нейронов равен  $I = I_0 = 5\xi$ ,  $\alpha, \beta, \xi$  - случайные числа от 0 до 1 (распределение равномерное). Тормозные нейроны имеют следующие значения параметров:  $a = 0.02 + 0.08\gamma$ ,  $b = 0.25 - 0.05\delta$ ,  $c = -65$ ,  $d = 2$  и внешний ток в отсутствие других токов от других нейронов равен  $I = I_0 = 2\zeta$ , где  $\gamma, \delta, \zeta$  - случайные числа от 0 до 1. В качестве начальных условий используются значения  $v(0) = -65$ ,  $u(0) = bv(0)$ .
  - Нейронная сеть может быть смоделирована с помощью полного графа. Матрица смежности  $\mathbf{W}$  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  $i$  одновременно увеличивается на величину  $W_{ij}$  и затем сразу же падает до нуля, что и моделирует передачу импульса по нейронной сети. Значение  $W_{ij}$  равно  $0.5\theta$ , если нейрон  $j$  является возбуждающим, и  $-\tau$ , если тормозным, где  $\theta, \tau$  - 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 сети.

## 2 Цель выполнения лабораторной работы

Цель выполнения лабораторной работы – изучение методов решения задачи Коши для систем ОДУ 1-го порядка. Программная реализация рассматриваемых методов, результатом работы которой являются функции для построения дискретной траектории ОДУ. Применение разработанных программ для моделирования работы биологического нейрона и нейронной сети. Анализ полученных графиков и анализ рассматриваемых методов на основе этих графиков.

## 3 Разработка функций.

Каждая из функций, которые необходимо реализовать, возвращает дискретную траекторию системы ОДУ из постановки задачи. Таким образом, имеет смысл разработать набор функций, реализующих методы решения системы ОДУ произвольной размерности. Такая система ОДУ выглядит следующим образом:

$$\frac{d}{dt} \begin{bmatrix} y_1(t) \\ y_2(t) \\ \vdots \\ y_n(t) \end{bmatrix} = \begin{bmatrix} f_1(t, y_1, \dots, y_n) \\ f_2(t, y_1, \dots, y_n) \\ \vdots \\ f_n(t, y_1, \dots, y_n) \end{bmatrix},$$

или кратко в векторном виде:

$$\frac{d\mathbf{y}}{dt} = \mathbf{f}(t, \mathbf{y}),$$

где  $y_1(a) = \alpha_1, \dots, y_n(a) = \alpha_n, t \in [a, b]$ .

Очевидно, что разработанные функции должны иметь одинаковую сигнатуру, и различные ключевые детали реализации, которые зависят от метода решения задачи Коши в конкретной функции. В нашем случае это метод Эйлера, неявный метод Эйлера и метод Рунге-Кутты 4-го порядка. Сигнатура этих функций несколько отлична от той, что дана в постановке задачи (для удобства и большей гибкости использования), но сохраняет ту же суть. Общий вид описанных функций представлен в листинге 1.

Листинг 1. Общий вид функции для решения задачи Коши и построения дискретной траектории системы ОДУ.

---

```
1 def method(x_0, start_time, end_time, func, step,
2     additional_condition=default_additional_condition):
3     t_nodes = np.arange(start_time, end_time, step=step)
4     w_nodes = [additional_condition(np.array(x_0))]
5
6     # Code, depends Cauchy problem solution method
7
8     return t_nodes, np.array(w_nodes).transpose()
```

---

Общая структура функции, представленная в листинге 1 позволяет строить траекторию для системы ОДУ произвольной размерности. Особое внимание следует уделить параметру функции `additional_condition`. Данный параметр принимает функцию, которая производит предварительную обработку вычисленных значений точек траектории. В контексте данного задания, этот параметр позволяет соблюдать ограничения (2) на значения  $v$  и  $u$ . Подробная информация о параметрах реализованных функций содержится в строках документации реализованных функций в исходных кодах программы в файле `ODE_solver_methods.py`.

### Реализация функции `euler`.

Данная функция решает задачу Коши при помощи метода Эйлера. Формулировка метода Эйлера в многомерном случае имеет следующий вид:

$$\mathbf{w}_0 = \alpha,$$

$$\mathbf{w}_{i+1} = \mathbf{w}_i + h\mathbf{f}(t_i, \mathbf{w}_i),$$

где  $\alpha$  - многомерная точка дискретных данных.  $t_i \in [a; b]$ ,  $t_i = a + ih, i = 1, \dots, m$ , а  $h = \frac{b-a}{m} = t_{i+1} - t_i$ .  $m$  - количество дискретных точек  $t_i$ .  $\mathbf{f}(t_i, \mathbf{w}_i)$  - функция из правой части нормализованного ОДУ, то есть разрешенного относительно производной.

Таким образом, этот метод можно реализовать при помощи цикла, представленного в листинге 2. На каждой итерации этого цикла вычисляется очередное значение  $w_i$  в момент времени  $t_i$ , таким образом строится дискретная траектория системы ОДУ.

Листинг 2. Реализация метода Эйлера для определения траектории системы ОДУ.

---

```

1 def w_next(w_prev, t_node):
2     return w_prev + step * func(w_prev, t_node)
3
4 for i, t in enumerate(t_nodes[1:]):
5     w_nodes.append(additional_condition(w_next(w_nodes[i], t)))

```

---

### Реализация функции `implicit_euler`.

Метод Эйлера можно представить, как  $p$ -шаговый метод Адамса-Башфорта при  $p = 1$ . То есть метод Адамса-Башфорта задает явную формулировку  $(p - 1)$ -шагового метода. Как известно из лекций, неявную формулировку  $(p - 1)$ -шагового метода задает метод Адамса-Моултона. Если принять  $p = 1$ , то очевидно, получим выражение метода Эйлера в неявном виде, которое выглядит следующим образом:

$$\mathbf{w}_0 = \alpha,$$

$$\mathbf{w}_{i+1} = \mathbf{w}_i + h\mathbf{f}(t_i, \mathbf{w}_{i+1}),$$

где  $\alpha$  - многомерная точка дискретных данных.  $t_i \in [a; b]$ ,  $t_i = a + ih, i = 1, \dots, m$ , а  $h = \frac{b-a}{m} = t_{i+1} - t_i$ .  $m$  - количество дискретных точек  $t_i$ .  $\mathbf{f}(t_i, \mathbf{w}_i)$  - функция из правой части нормализованного ОДУ.

Таким образом,  $\mathbf{w}_{i+1}$  будет корнем приведенного выражения. Следует учесть, что в общем случае система уравнений, которую придется решить, то есть

$$0 = \mathbf{w}_i + h\mathbf{f}(t_i, \mathbf{w}_{i+1}) - \mathbf{w}_{i+1},$$

будет нелинейной. Поэтому для реализации этого метода воспользуемся вспомогательными программными средствами для решения нелинейных уравнений, а именно функцией `root` из python пакета `scipy.optimize`.

Суть реализации функции для этого метода остается схожей с реализацией для явного метода Эйлера, а именно итерационный обход всех значений  $t_i$  и поиск дискретной траектории системы ОДУ. Однако теперь на каждой итерации будет решаться система уравнений приведенная выше, как показано в листинге 3.

Листинг 3. Реализация неявный метода Эйлера для определения траектории системы ОДУ.

---

```

1 for i, t in enumerate(t_nodes[1:]):
2     solution = root(
3         lambda w_next: w_nodes[i] + step * func(w_next, t) - w_next,
4         w_nodes[i],
5     )
6     w_nodes.append(additional_condition(solution.x))

```

---

### Реализация функции `runge_kutta`.

В этом случае для решения задачи Коши используется метод Рунге-Кутты 4-го порядка. Его формулировка для систем ОДУ приводится в лекциях следующим образом:

$$\begin{aligned}
 \mathbf{w}_0 &= \alpha, \\
 \mathbf{k}_1 &= h\mathbf{f}(t_i, \mathbf{w}_i), \\
 \mathbf{k}_2 &= h\mathbf{f}\left(t_i + \frac{h}{2}, \mathbf{w}_i + \frac{1}{2}\mathbf{k}_1\right), \\
 \mathbf{k}_3 &= h\mathbf{f}\left(t_i + \frac{h}{2}, \mathbf{w}_i + \frac{1}{2}\mathbf{k}_2\right), \\
 \mathbf{k}_4 &= h\mathbf{f}(t_i + h, \mathbf{w}_i + \mathbf{k}_3), \\
 \mathbf{w}_{i+1} &= \mathbf{w}_i + \frac{1}{6}(\mathbf{k}_1 + 2\mathbf{k}_2 + 2\mathbf{k}_3 + \mathbf{k}_4), \quad i = 0, 1, \dots, m-1,
 \end{aligned}$$

где  $\alpha$  - многомерная точка дискретных данных.  $t_i \in [a; b]$ ,  $t_i = a + ih, i = 1, \dots, m$ , а  $h = \frac{b-a}{m} = t_{i+1} - t_i$ .  $m$  - количество дискретных точек  $t_i$ .  $\mathbf{f}(t_i, \mathbf{w}_i)$  - функция из правой части нормализованного ОДУ.

Таким образом, снова возвращаемся к циклу, который в каждой итерации вычисляет дискретную траекторию системы ОДУ. Однако необходимо также реализовать функции для вычисления  $\mathbf{k}_1, \mathbf{k}_2, \mathbf{k}_3, \mathbf{k}_4$ . Реализация этого метода представлена в листинге 4.

Листинг 4. Реализация метода Рунге-Кутты 4-го порядка для определения траектории системы ОДУ.

---

```
1 def k_1(w_prev, t_node):
2     return step * func(w_prev, t_node)
3
4 def k_2(w_prev, t_node):
5     return step * func(w_prev + 0.5 * k_1(w_prev, t_node), t_node + step * 0.5)
6
7 def k_3(w_prev, t_node):
8     return step * func(w_prev + 0.5 * k_2(w_prev, t_node), t_node + step * 0.5)
9
10 def k_4(w_prev, t_node):
11     return step * func(w_prev + k_3(w_prev, t_node), t_node + step)
12
13 for i, t in enumerate(t_nodes[1:]):
14     w = w_nodes[i]
15     w_next = w + (1 / 6) * (k_1(w, t) + 2 * k_2(w, t) + 2 * k_3(w, t) + k_4(w, t))
16     w_nodes.append(additional_condition(w_next))
```

---

#### 4 Расчет траекторий динамической системы при использовании разных характерных режимов. Построение графиков полученных траекторий.

Применим функции, которые были реализованы в предыдущем пункте, к различным конфигурациям выражения (1), (2), параметризуя их в зависимости от характерных режимов из таблицы 1. В результате получим набор дискретных траекторий, которые соответствуют характерным режимам работы нейрона. Графики этих траектории в режиме работы Tonic Spiking отображен на рисунке 1, в режиме Phasic Spiking на рисунке 2, в режиме Chattering на рисунке 3 и в режиме Fast Spiking на рисунке 4.



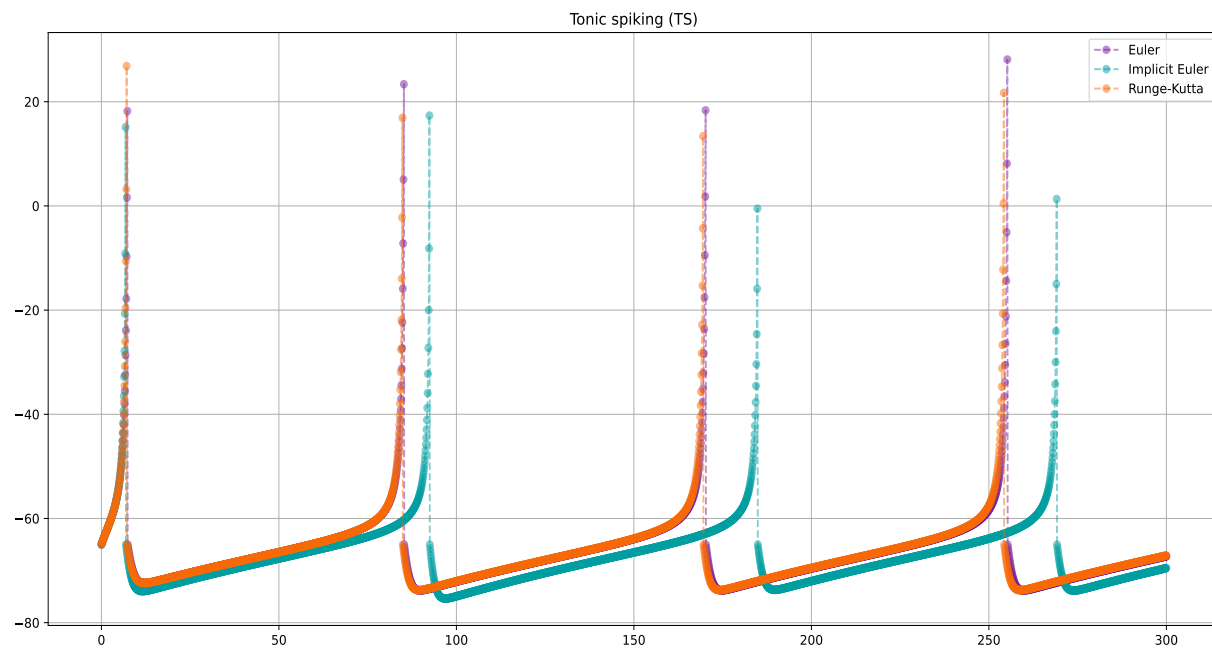


Рис. 1. Работа нейрона в режиме tonic spiking.

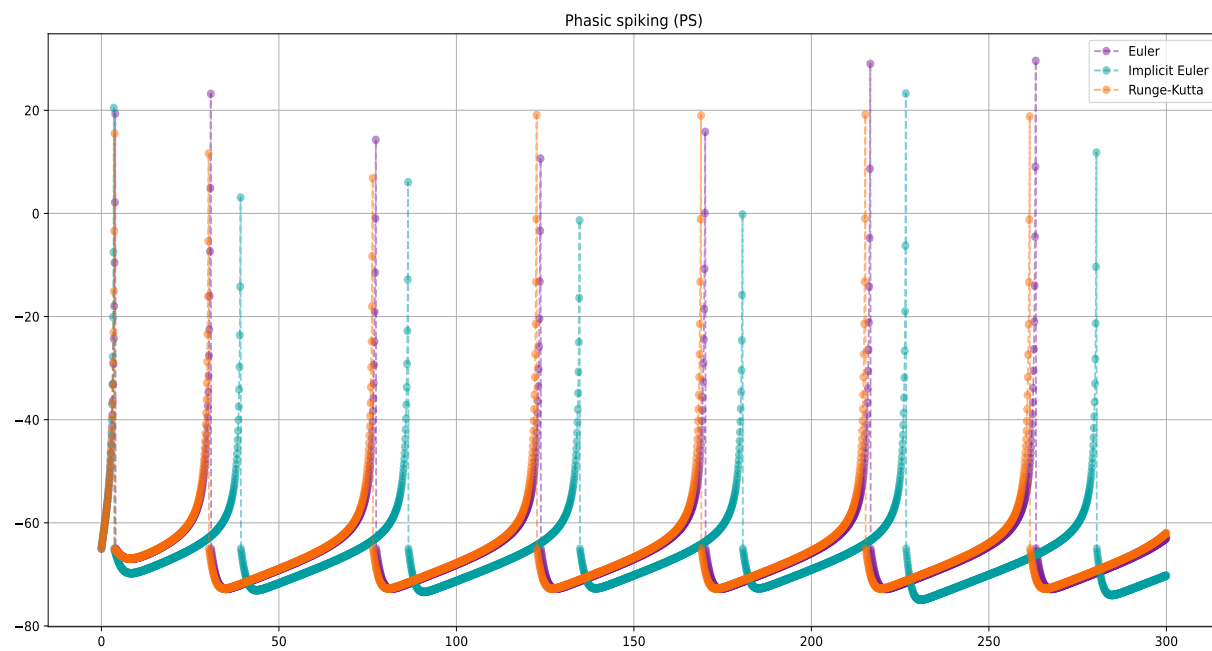


Рис. 2. Работа нейрона в режиме phasic spiking.

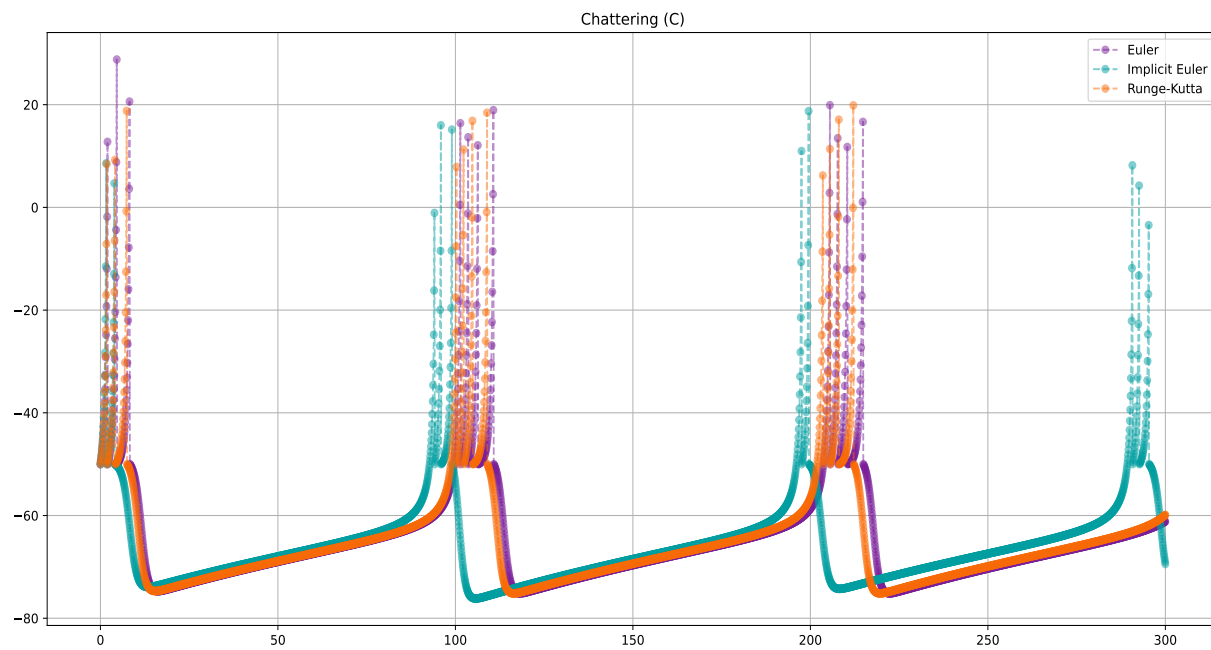


Рис. 3. Работа нейрона в режиме chattering.

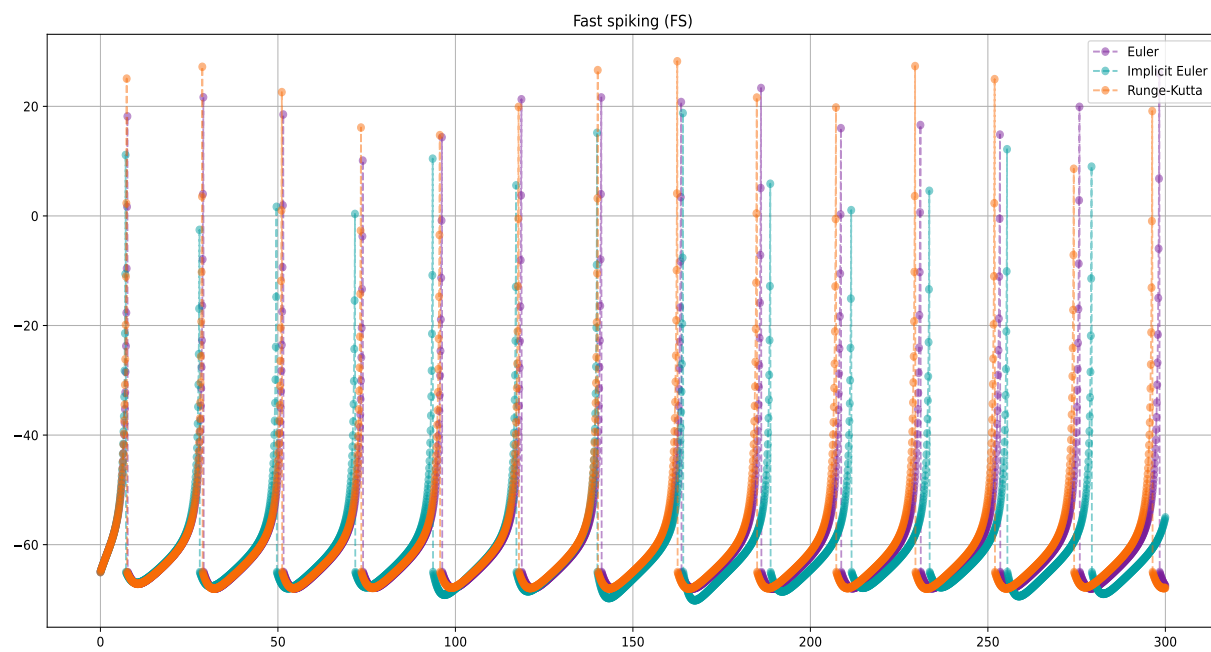


Рис. 4. Работа нейрона в режиме fast spiking.

## 5 Анализ построенных графиков.

Проанализируем полученные графики. Рассмотрим график работы модели нейрона в режиме tonic spiking (Рис. 1). Первый всплеск возникает практически сразу при воздействии тока на нейрон, а затем всплески происходят с примерно одинаковой периодичностью. Это явление в нейробиологии называют процессом адаптации нейрона. Так как чувствительность  $u$  к флуктуациям разности потенциалов достаточно мала, всплески происходят довольно редко.

В отличие от tonic spiking, режим работы phasic spiking (Рис. 2) выделяется тем, что всплесков, происходящих в результате адаптации нейрона, больше (а именно, на графике можно заметить два всплеска адаптации). Кроме того, частота появления всплеска стала значительно выше. Это связано с тем, что параметр  $b$ , то есть чувствительно переменной восстановления мембраны  $u$  к флуктуациям, больше, чем тот же параметр у tonic spiking.

Нейрон, в режиме работы chattering (Рис. 3), имеет характерную особенность. Можно наблюдать сразу несколько всплесков подряд. Такое явление связано с высоким значением потенциала мембраны сразу после импульса.

Особенность режима работы fast spiking, заключается в его высокой скорости восстановления мембраны после импульса. Параметр  $a$  для этого режима, больше, чем у других режимов, рассматриваемых в этой работе.

## 6 Анализ методов решения задачи Коши для систем ОДУ.

Основной интерес в данной работе представляет анализ рассматриваемых в ней методов решения задачи Коши для систем ОДУ.

Общей чертой всех рассмотренных методов является их рекуррентная форма. То есть, чтобы построить дискретную траекторию, необходимо итеративно вычислять значения новых точек траектории, опираясь на уже найденные.

Однако, особое внимание следует уделить принципиальным отличиям данных методов.

При помощи метода Эйлера можно производить вычисления быстрее, чем другими методами. При этом необходимо помнить, что его локальная погрешность имеет порядок точности  $O(h^2)$ . Оценка глобальной погрешности учитывает, что локальная погрешность накапливается при вычислении  $m$  точек дискретной траектории, поэтому она имеет порядок точности  $mO(h^2) = O(h)$ . Таким образом, метод Эйлера имеет самый низкий порядок точности среди других рассматриваемых методов.

В отличие от метода Эйлера, метод Рунге-Кутты 4-го порядка имеет меньшую погрешность вычислений. Следует уточнить с чем это связано. Получение рекуррентной формулы для решения задачи Коши, предполагает предварительное разложение функции из ОДУ в ряд Тейлора. В случае метода Эйлера функцию раскладывают в ряд Тейлора первого порядка. Для получения выражения метода Рунге-Кутты  $n$ -го порядка, функцию раскладывают в ряд Тейлора  $n$ -го порядка. В результате аппроксимации этого разложения, как известно из лекций, можно получить искомое выражение. Таким образом данная аппроксимация будет точной вплоть до члена  $O(h^n)$ ,

в рассматриваемом случае вплоть до  $O(h^4)$ . Однако, следует учитывать, что данный метод, при вычислении системы ОДУ 1-го порядка в ходе итеративного вычисления значений описанной рекуррентной формулы, требуется вычислить функцию  $\mathbf{f}(t, \mathbf{y})$  4 раза для одной итерации, что увеличивает сложности вычисления.

Выражение неявного метода Эйлера, как уже говорилось ранее, можно получить из формулы метода Адамса-Моултона, при подстановке шага  $p = 1$ . В лекции мы обсуждали, что  $(p - 1)$ -шаговый метод Адамса-Моултона имеет порядок точности  $O(h^{p+1})$ , в случае неявного метода Эйлера, глобальная погрешность будет иметь порядок  $O(h^2)$ . В таком случае порядок точности неявного метода Эйлера выше, чем у явного. Однако, следует учитывать, что при использовании данного способа, необходимо будет решить, в общем случае, систему нелинейных уравнений, что является нетривиальной задачей и, как правило, требует больших вычислительных затрат. Кроме того, не стоит забывать о том, что численные методы решения нелинейных уравнений имеют собственную погрешность вычислений, которая складывается с погрешностью неявного метода Эйлера. Кроме того, неявный метод Эйлера, это единственный из рассматриваемых методов, который является абсолютно устойчивым. Что, как известно из лекций значит, что вне зависимости от шага  $h$ , неявный метод Эйлера, примененный к устойчивому ОДУ также будет устойчивым.

## 7 Моделирование работы нейронной сети.

Для моделирования работы нейронной сети из поставленной задачи была реализована программа. Описываемая программа создает массив значений  $v$  и  $u$  для каждого из 1000 нейронов (800 возбуждающих и 200 тормозных), этот массив сохраняет в себе значения потенциалов мембраны и переменных восстановления мембраны в конкретный момент времени  $t_i$  для всех нейронов сети. Задаются параметры модели для каждого нейрона и начальные условия задачи Коши. Создается матрица смежности, которая заполняется значениями в соответствии с заданием. Далее итеративно по времени (время моделирования составляет 1000 мс) для каждого нейрона была рассчитана дискретная траектория при помощи метода Эйлера и зафиксированы их всплески в определенные моменты времени. В итоге получилось построить зависимость (Рис. 5) всплесков отдельных нейронов от времени, на которой отчетливо видны колебания нейронов сети.

Можно заметить, что колебания нейронов сети имеют частично синхронный характер. Под частичной синхронностью подразумевается наличие эпизодов, когда импульс в нейронах происходит практически одновременно. Если обратить внимание на график, то эти эпизоды выглядят как темные полосы, легко заметные среди прочих импульсов. Можно сделать вывод, что хотя нейроны в сети и смоделированы случайным образом, постепенно они организуются и демонстрируют ритмическое поведение. Частоту синхронных импульсов, которую мы наблюдаем, называют альфа-частотой, и она соответствует, 10 Гц. Эту частоту в нейробиологии также называют альфа-ритм. Как описывает Е. Ижикевич в своих работах, этот ритм соответствует активности головного мозга млекопитающего в состоянии бодрствования.

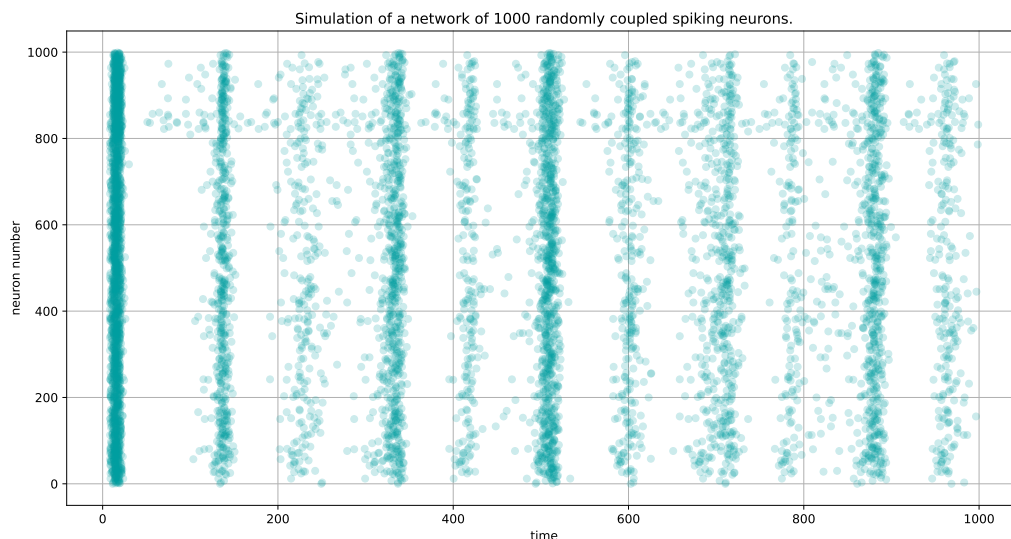


Рис. 5. Импульсы модели нейронов сети, возникающие в течении 1000мс.

## 1 Заключение

1. В ходе лабораторной работы рассмотрены методы решения задачи Коши для ОДУ 1-го порядка.
2. На основе рассмотренных методов реализованы функции, каждая из которых возвращает траекторию ОДУ.
3. При помощи разработанных функций, проведено моделирование работы биологического нейрона, из которого можно заключить, что в зависимости от различных параметров модели, выходные графики могут существенно отличаться.
4. Проведен анализ методов решения задачи Коши для ОДУ 1-го порядка.
5. Проведено моделирование нейронной сети, состоящий из 1000 нейронов. Анализ сети показал, что импульсы нейронов, параметры модели которых подбираются случайным образом, имеют частично синхронный характер.



### Список использованных источников

1. Першин А.Ю. Лекции по курсу «Вычислительная математика». Москва, 2018-2021. С. 140.
2. Першин А. Ю., Соколов А. П. Сборник постановок задач на лабораторные работы по курсу "Вычислительная математика". Москва, 2018-2021.

3. Eugenie M. Izhekevich Simple model of spiking neurons. IEEE Transaction on neural networks, 2003.
4. William H. Press, Saul A. Teukolsky, William T. Vetterling Brian P. Flannery Numeric recipes. The art of scientific computing. Cambridge, 1988-2007.
5. Калиткин Н. Н. Численные методы. Издательство "Наука Москва-1978.

## Выходные данные

Дунайцев А. И.. Отчет о выполнении лабораторной работы по дисциплине «Вычислительная математика». [Электронный ресурс] — Москва: 2021. — 14 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 РК6-54Б, Дунайцев А. И.

2021, осенний семестр