

МИНОБРНАУКИ РОССИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н.Г.ЧЕРНЫШЕВСКОГО»

Кафедра физики открытых систем

**Создание сайта для моделирования сетей фазовых осцилляторов
Курамото с различными типами связи**

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

Студента(ки) 4 курса 4041 группы
направления (специальности) 09.03.02. Информационные системы и
технологии

код и наименование направления (специальности)
института физики

наименование факультета, института, колледжа
Полякова Сергея Сергеевича

фамилия, имя, отчество

Научный руководитель
профессор кафедры ФОС,

д.ф.-м.н., доцент
должность, уч. степень, уч. звание

дата, подпись

О.И. Москаленко
инициалы, фамилия

Заведующий кафедрой

д.ф.-м.н., профессор
должность, уч. степень, уч. звание

дата, подпись

А.А. Короновский
инициалы, фамилия

Саратов 2022 год

Введение

В последние годы веб-приложения в интернете стали популярными среди пользователей и в сфере разработки. Все чаще можно увидеть, что многие авторы популярных программных пакетов отказываются от написания отдельных программ под конкретные ОС, а пишут веб-приложения. К тому же, веб-приложение будет доступно на всех устройствах, где есть веб-браузер и доступ в интернет. В данном контексте использование веб-технологий приобретает высокую актуальность [1].

Другим востребованным в настоящее время направлением деятельности является моделирование сложных систем с большим числом параметров. Численное моделирование позволяет провести изучение динамической системы и получить выводы об особенностях поведения данной модели при различных условиях. Модель фазовых осцилляторов Курамото является примером такой системы [2]. Это одна из наиболее представительных моделей связанных фазовых осцилляторов, в которой могут наблюдаться режимы синхронизации – одного из фундаментальных явлений природы. Синхронизацию можно рассматривать как явление самоорганизации различных взаимодействующих систем.

Целью данной бакалаврской работы является создание веб-сайта, который позволяет производить численное моделирование модели Курамото интерактивно в режиме реального времени. Поскольку такого сайта на момент написания работы в сети найдено не было, его создание целесообразно. Существование такого сайта дает возможность пользователям без написания программ исследовать поведение заданной модели.

В качестве объектов исследования были выбраны классическая и адаптивная модели Курамото. Для выполнения работы был использован язык программирования Python и веб-фреймворк Django.

Описание работы

Основная часть работы состоит из трех глав.

В первой главе описываются виды синхронизации в моделях Курамото, дальнейшему рассмотрению которых посвящены последующие главы.

Модель Курамото изначально была основана на феномене коллективной синхронизации, при котором огромная система осцилляторов самопроизвольно замыкается на общую частоту, несмотря на неизбежные различия в собственных частотах отдельных осцилляторов.

Классическая модель Курамото имеет вид:

$$\dot{\theta}_i = \omega_i + \frac{K}{N} \sum_{j=1}^N \sin(\theta_j - \theta_i), \quad i = 1, \dots, N \quad (1)$$

где θ_i – фаза осциллятора i , ω_i – собственная частота осциллятора i , N – общее число осцилляторов в системе, а K – константа, называемая параметром связи.

Параметр порядка характеризует коллективный ритм, создаваемый всей системой:

$$r e^{i\psi} = \frac{1}{N} \sum_{j=1}^N e^{i\theta_j} \quad (2)$$

Другое обобщение модели Курамото связано с изучением эффектов и явлений в адаптивных осцилляторных сетях [3]. Особенностью таких сетей является наличие свойства адаптивности межэлементных взаимодействий, которое заключается в динамическом изменении силы связи между элементами сети в зависимости от их текущих состояний. Для описания динамики элементов в сетях с таким сложным нестационарным характером взаимодействий используют обобщение модели (1) следующего вида:

$$\dot{\theta}_i = \omega_i + \frac{K|\omega_i|}{N} \sum_{j=1}^N \sin(\theta_j - \theta_i), \quad i = 1, \dots, N. \quad (3)$$

Здесь θ_i и ω_i – по-прежнему фаза и собственная частота каждого осциллятора, K – параметр связи.

Моделирование классической и адаптивной моделей Курамото осуществлялось на основе языка программирования Python.

Была написана программа, которая позволяет получить решение системы большого количества взаимосвязанных дифференциальных уравнений. Проведены следующие построения:

Для классической модели:

- зависимостей сигналов от времени для различных значений параметра связи;
- зависимости параметра порядка от параметра связи.

Для адаптивной модели:

- зависимости параметра порядка от параметра связи.

Для классической модели были выбраны следующие значения параметров для системы (1): число осцилляторов $N = 100$, параметр связи $K = 5$, промежуток времени $[0, 1]$, частоты осцилляторов ω_i задавались случайно из диапазона $[1;3]$ или $[2.99, 3.1]$, начальные условия осцилляторов задавались случайно из диапазона $[0, 2\pi]$.

В результате выполнения программы выполнялось построение двух графиков: зависимости сигналов $x_i(t) = \sin(\omega_i(t) + \theta_0)$ от времени, и зависимости параметра порядка от времени.

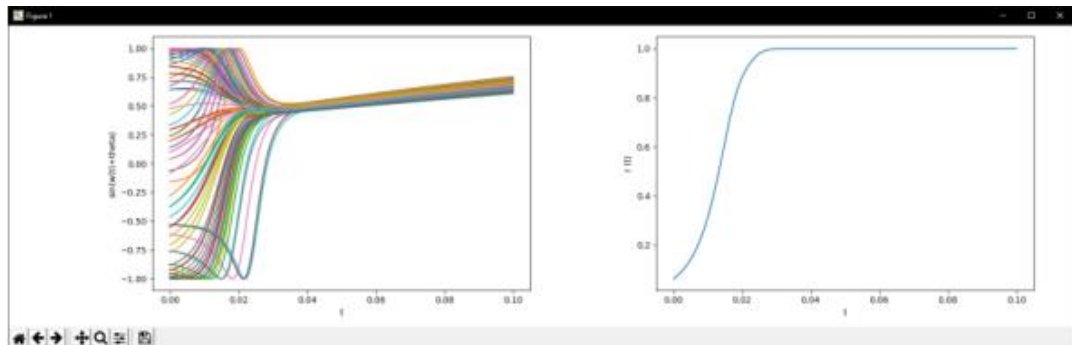


Рисунок 1 – Полная синхронизация в системе (1) при $\omega_i \in [1;3]$, значение параметра связи $K = 5$

На рисунке 1 заметно, как фазы осцилляторов быстро приобретают общую частоту и фазу. Осцилляторы синхронизовались. Параметр r приобретает постоянное значение, становясь равным единице, что также говорит о достижении синхронизации [4].

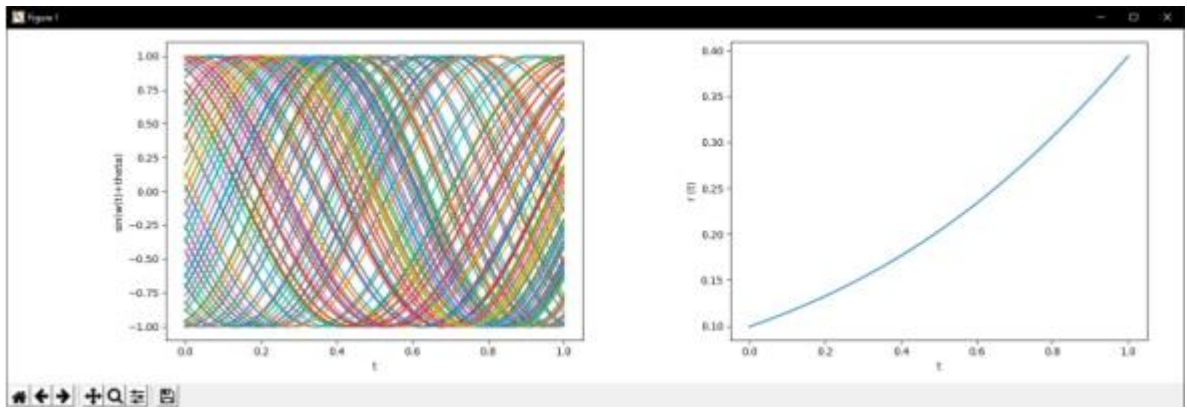


Рисунок 2 – Отсутствие синхронизации в системе (1) при $\omega_i \in [2.99; 3.1]$ при малом значении параметра связи

На рисунке 2 синхронизация системы отсутствует. Параметр порядка не достигает постоянного значения, что говорит о том, что осцилляторы колеблются несинхронно.

Для построения зависимости параметра порядка от параметра связи предыдущая программа была модифицирована с целью использования возможности изменения параметра связи.

На рисунке 3 представлены зависимости параметра порядка от параметра связи для сетей, состоящих из различного количества осцилляторов.

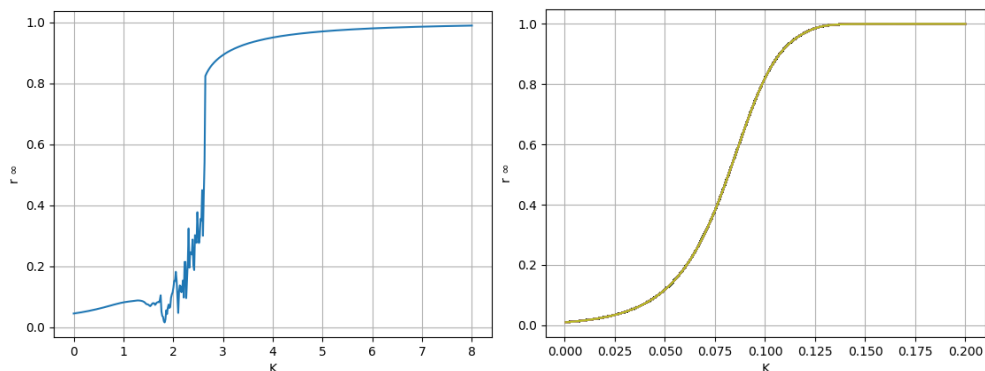


Рисунок 3 – Зависимости параметра порядка от параметра связи для сети из $N=1000$ осцилляторов, частоты распределены случайно в диапазоне $[1, 5]$ (слева), и для сети из $N=100$ осцилляторов с постоянной частотой $\omega = 2.99$ (справа)

Как можно заметить из приведенных выше рисунков, число осцилляторов в системе не сильно влияет на зависимость параметра порядка от параметра связи, но, при постоянной частоте синхронизация возникает значительно быстрее, а зависимость параметра порядка от параметра связи получается более плавной.

Для адаптивной модели в качестве начальных условий было выбрано треугольное распределение частот и равномерное распределение фаз. На рисунке 4 приведены зависимости параметра порядка от параметра связи для исследуемой модели при изменении параметра связи в прямом и обратном направлениях.

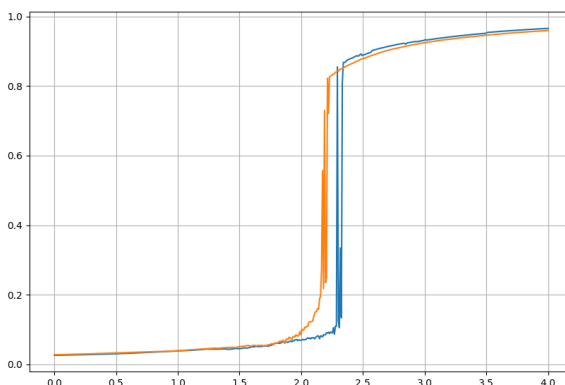


Рисунок 4 – Сравнение зависимости параметра порядка от параметра связи для сети из $N=5000$ осцилляторов при использовании адаптивной модели.

Из сравнения рисунков видно, что наблюдается образование подобия гистерезиса при изменении параметра связи в двух различных направлениях. Полученные путем численного моделирования зависимости оказались аналогичными графикам, приведенным в статьях [4],[6],[7].

Во второй главе рассматриваются технологии, использованные при создании сайта. Изначально для моделирования системы Курамото были написаны программы на языке программирования Python, предназначенные для выполнения на локальном компьютере. Для разработки веб-приложения этот код было необходимо адаптировать для работы в интернете. Для этого

использовался веб-фреймворк Django [8]. В качестве хостинга был выбран Heroku: он имеет бесплатный тариф и работает с Django приложениями [9]. Веб-приложения, обрабатывающие входящие HTTP-запросы параллельно, используют выделяемые ресурсы Heroku гораздо эффективнее, чем веб-приложения, обрабатывающие только один запрос за раз. Веб-фреймворк Django имеет удобные встроенные веб-серверы, но эти серверы обрабатывают только один запрос за раз. Поэтому был использован веб-сервер Gunicorn, а для кеширования файлов – модуль Whitenoise [10]. Whitenoise управляет кешированием и сжатием файлов для увеличения скорости их подачи на страницу и загрузки страницы. В результате получается работоспособная веб-страница с CSS, изображениями и текстом. Формулы можно писать с помощью LaTeX, используя плагин MathJax, который преобразовывает LaTeX код в странице в отображение в виде изображения. Для отображения графиков зависимости применяется mpld3 [11]. Библиотека mpld3 конвертирует результат построения из Matplotlib в JSON, встраиваемый в контейнер на сайте, в результате на странице появляется интерактивное окно с графиком. Для передачи и ввода данных используется стандартный метод запроса POST.

В третьей главе рассматривается сайт, созданный в ходе данной работы. Все элементы сайта вынесены на главную страницу в виде различных разделов (рисунок 5):

- Описание модели Курамото
- График зависимости сигналов от времени
- График зависимости параметра порядка от времени
- Описание и построение адаптивной модели

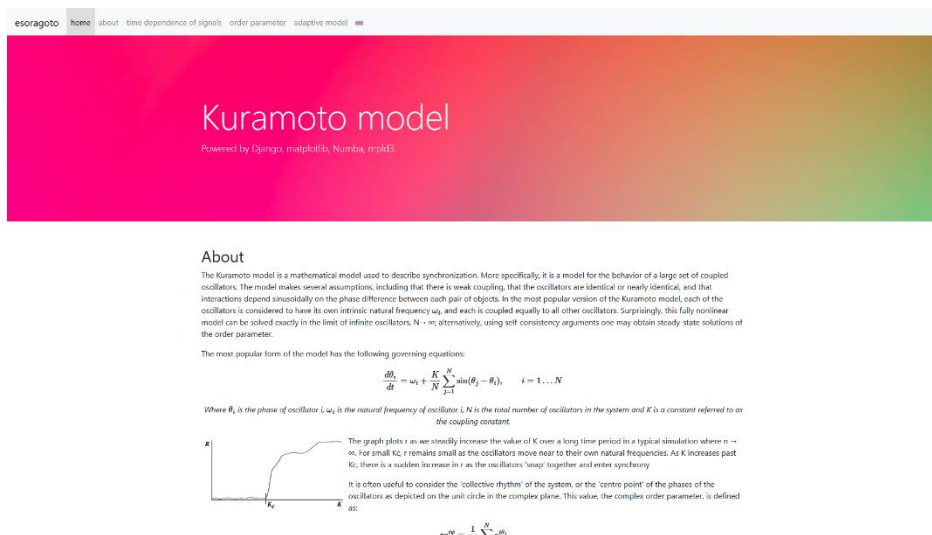


Рисунок 5 – Главная страница сайта

К каждому разделу можно сразу перейти, нажав на название раздела в шапке сайта. Уравнения написаны на LaTeX и выводятся с помощью MathJax. Есть возможность смены языка (рисунок 6).

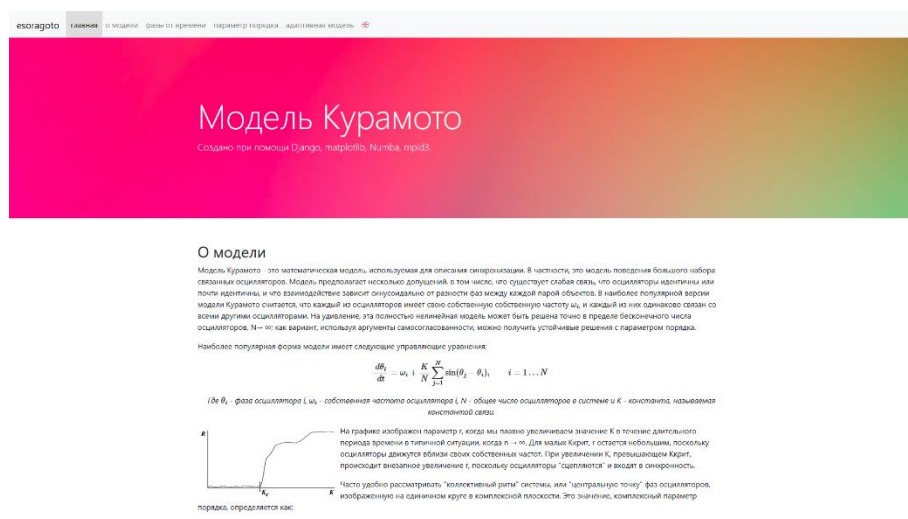


Рисунок 6 – Смена языка

Сайт доступен по следующему адресу: <https://esoragoto.herokuapp.com/>. Исходный код сайта был сделан свободно распространяемым и доступен в следующих репозиториях:

- <https://gitlab.com/icosane/esoragoto>
- <https://github.com/icosane/esoragoto>

Также реализована возможность на основе исходного кода с помощью кнопки “Deploy to Heroku” развернуть свой собственный экземпляр сайта на хостинге Heroku.

В случае графика зависимости параметра порядка от параметра связи сервер не всегда может стабильно выдержать вычисления при больших значениях. Поэтому была добавлена JavaScript галерея графиков с разным числом осцилляторов (рисунок 7).

Поскольку сервер, к сожалению, не может справиться с большими вычислениями и, скорее всего, упадет при большом количестве осцилляторов, вот несколько предварительно скомпилированных графиков для 10, 50, 100, 500 и 1000 осцилляторов. Шаг параметра связи равен 500.

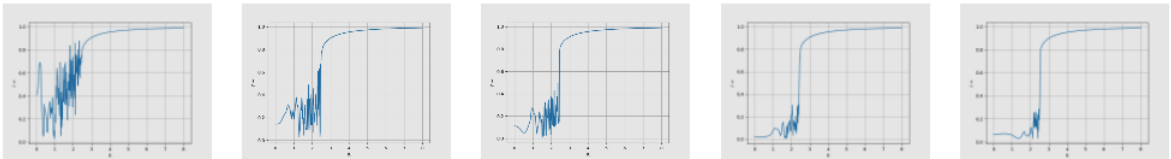


Рисунок 7 – Галерея графиков зависимости параметра порядка от параметра связи для различного числа осцилляторов

При нажатии на график можно увеличить изображение и просмотреть галерею (рисунок 8).

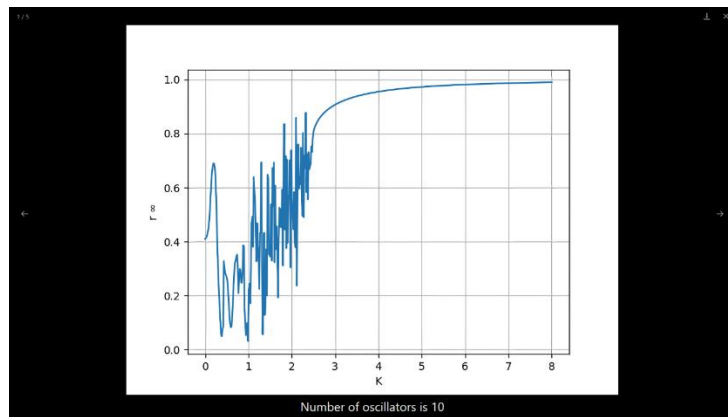


Рисунок 8 – Внутренний интерфейс галереи

Моделирование адаптивной модели (рисунок 9) является еще более ресурсоемким процессом, поэтому, несмотря на то что такая возможность есть, пользоваться ею нерационально из-за большого времени выполнения.

В связи с этим, предусмотрена возможность загрузить файл программы напрямую с сайта, чтобы выполнить построения на локальном устройстве. Ссылки для всех необходимых для этого дополнительных инструментов приведены в описании к загрузке (рисунок 10).

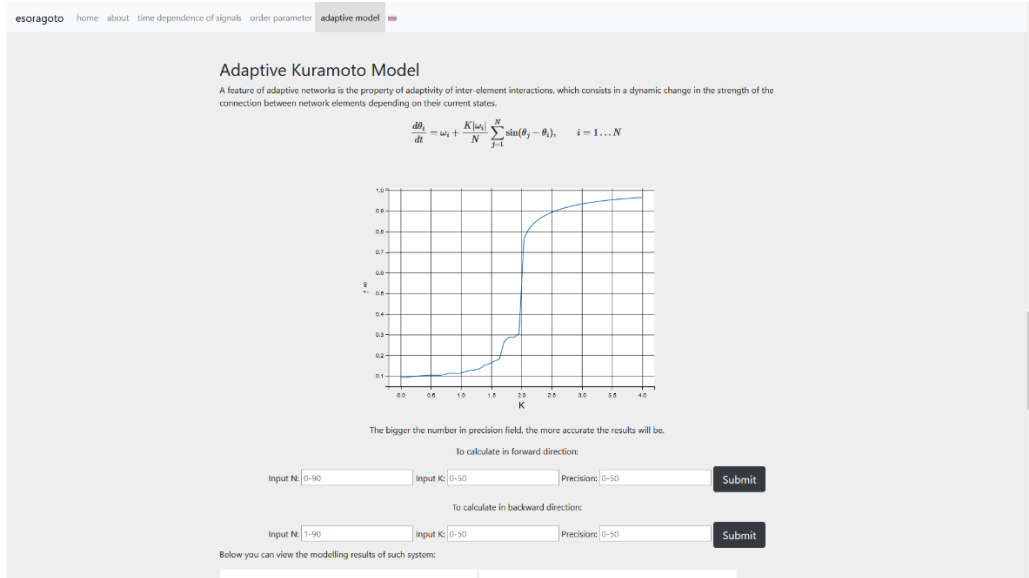


Рисунок 9 – Описание и график адаптивной модели

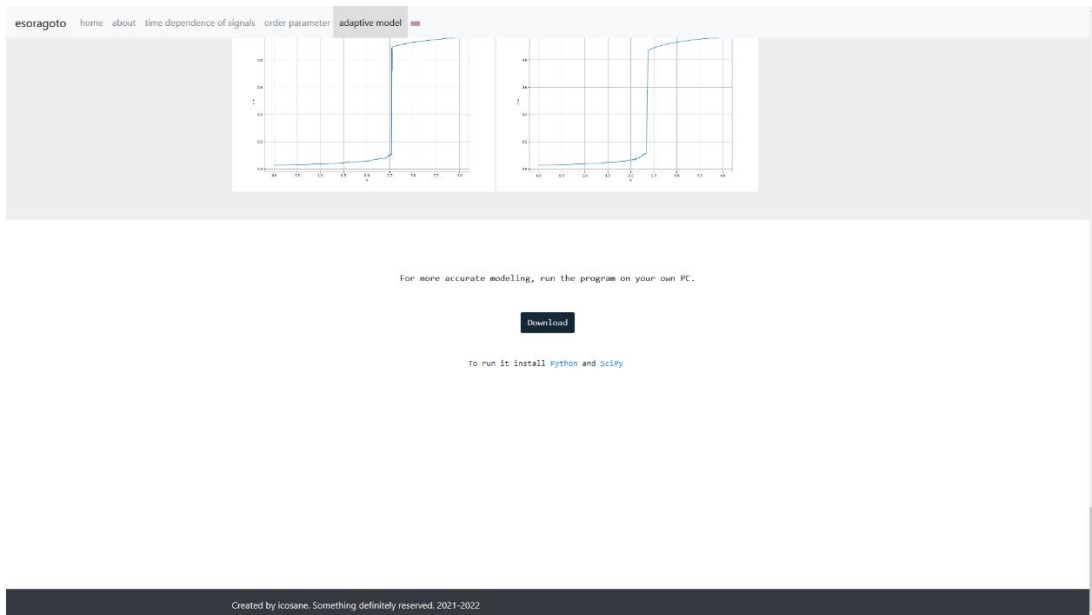


Рисунок 10 – Кнопка загрузки файла с программой

Заключение

В рамках бакалаврской работы был разработан веб-сайт, который позволяет интерактивно в режиме реального времени производить численное моделирование модели Курамото. В качестве объектов исследования были выбраны классическая и адаптивная модели Курамото. В ходе работы было произведено моделирование выбранных систем с использованием языка Python и получены графики, иллюстрирующие зависимости характеристик систем от различных значений параметра связи.

Полученные графики оказались идентичными графикам, приведенным в работах [2] [4] при аналитическом решении задачи, что подтверждает правильность проведенного численного моделирования.

Было выяснено, что скорость синхронизации системы зависит от величины параметра связи: чем больше данный параметр, тем меньше время наступления синхронизации между осцилляторами.

Для адаптивной модели наблюдалось подобие гистерезиса, описанного в работах [4] [6].

Разработан сайт, который дает возможность производить численное моделирование модели Курамото интерактивно в режиме реального времени. Исходный код сделан свободно распространяемым и доступен в следующих репозиториях:

- <https://gitlab.com/icosane/esoragoto>
- <https://github.com/icosane/esoragoto>

Список литературы

1. **“The future of the web and native apps”** [Электронный ресурс], URL: <https://ilyabirman.net/meanwhile/all/web-or-native-future/>, свободный доступ. — Загл. с экрана. — Яз. англ. (Дата обращения 02.05.2022)
2. **Strogatz S. H.**, From Kuramoto to Crawford: exploring the onset of synchronization in populations of coupled oscillators [Текст]/ S. H. Strogatz // *Physica D: Nonlinear Phenomena*, 143(1-4), 2000., pp.1-20.
3. **Касаткин, Д. В.**, Нелинейные явления в осцилляторных сетях Курамото с динамическими связями. [Текст] / Касаткин, Д. В., Емельянова, А. А., & Некоркин, В. И. // *Известия высших учебных заведений. Прикладная нелинейная динамика*, 29(4), 635-675.
4. **Hu, X.**, Exact solution for first-order synchronization transition in a generalized Kuramoto model [Текст] / Hu, X., Boccaletti, S., Huang, W., Zhang, X., Liu, Z., Guan, S., & Lai, C. H. // *Scientific reports*, 4(1), 1-6.
5. **Wordsworth J.**, “An Introduction to Coupled Oscillators.” [Электронный ресурс] / Wordsworth J. // *Dynamical Systems Tutorial*. URL: <http://tutorials.siam.org/dsweb/cotutorial/index.php>, свободный доступ. — Загл. с экрана. — Яз. англ. (Дата обращения 02.05.2022)
6. **Zhang. X.**, Explosive Synchronization in Adaptive and Multilayer Networks [Текст]/ X Zhang, S Boccaletti, S Guan, Z Liu // *Physical review letters*, 114(3), 038701.
7. **Danziger M.M.**, Explosive synchronization coexists with classical synchronization in the Kuramoto model [Текст]/ М М. Danziger, О. I. Moskalenko, S. A. Kurkin, X. Zhang, S. Havlin, S. Boccaletti // *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 26(6), 065307.
8. **“Django Web Framework (Python)”** [Электронный ресурс], URL: <https://developer.mozilla.org/en-US/docs/Learn/Server-side/Django>, свободный доступ. — Загл. с экрана. — Яз. англ. (Дата обращения 02.05.2022)

9. **“Getting Started on Heroku with Python”** [Электронный ресурс], URL: <https://devcenter.heroku.com/articles/getting-started-with-python>, свободный доступ. — Загл. с экрана. — Яз. англ. (Дата обращения 02.05.2022)
10. **“WhiteNoise”** [Электронный ресурс], URL: <http://whitenoise.evans.io/en/stable/>, свободный доступ. — Загл. с экрана. — Яз. англ. (Дата обращения 02.05.2022)
11. **“Bringing Matplotlib to the Browser ”** [Электронный ресурс], URL: <https://mpld3.github.io/>, свободный доступ. — Загл. с экрана. — Яз. англ. (Дата обращения 02.05.2022)