МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»

Университет ИТМО

дисциплина
Физика с элементами компьютерного моделирования

Отчет по проекту

«Модель Курамото»
Вариант №15

Выполнили:
студент гр. N3145
Ложкин Владислав Сергеевич,
студент гр. N3150
Диденко Дмитрий Эдуардович

Проверил:
Барышникова К.В.

Санкт-Петербург
2023

Цель работы:

Демонстрация динамического изменения частот при разных K (коэффициентах связи) в модели Курамото, визуализация поставленной задачи, поиск критических точек смены режима.

Исходные данные:

Данные поставленной задачи для визуализации (15 вариант):

Два бегуна бегут по стадиону. Изначальные их скорости были 10 км/ч и 12 км/ч, но один из них явно хочет бежать вместе со вторым, а второй равнодушен к этому (бежит со своей изначальной скоростью).

Задачи, решаемые при выполнении работы:

1. Написание программы, визуализирующей модель Курамото в общем случае, то есть в системе должна произойти синхронизация движения объектов.
2. Изменения программы под условие поставленной задачи.
3. Поиск критических точек смены режимов при разных K.

Ход работы:

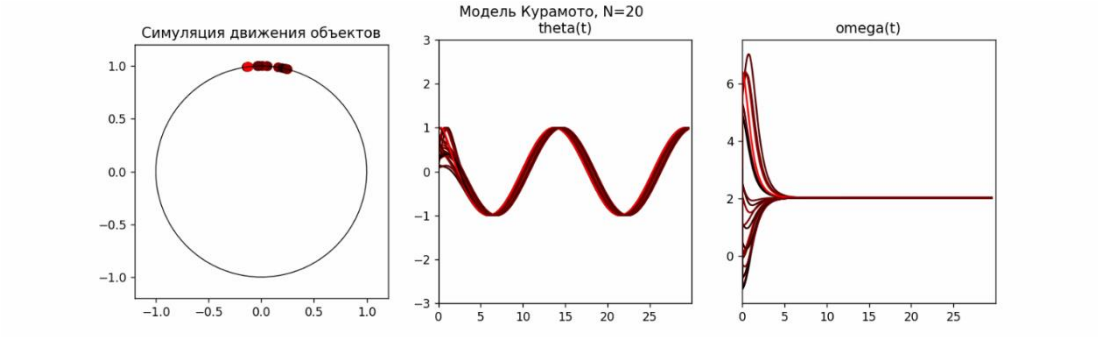
Начальные параметры объектов, такие как изначальная фаза, угловая частота, коэффициент связи выбираются случайным образом и заносятся на первые позиции в соответствующие массивы. Причем выбрано делать $K_{ij} = K_{ji}$. После этого в специальной функции просчитываются все последующие данные, такие как фаза и угловая частота, путем дифференцирования фазы по времени методом Эйлера. Формула для просчета берется из модели Курамото:

$$\dot{\varphi}_i = \omega_{oi} + \frac{1}{N}\sum_{j=1}^{N} K_{ij}\, sin(\varphi_j - \varphi_i)$$

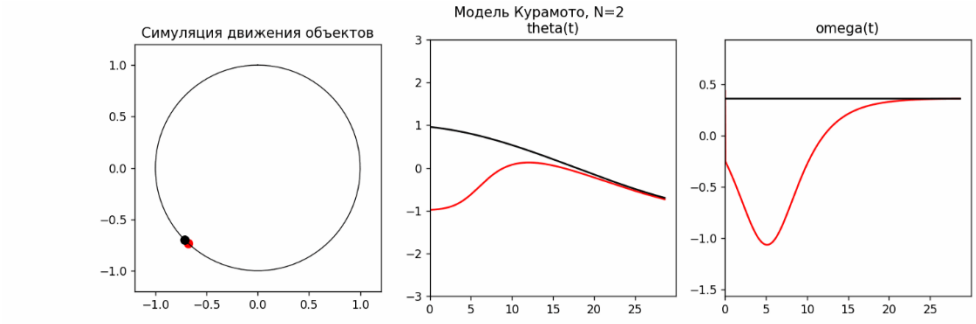
Дифференцирование происходит делением общего времени на маленькие промежутки, которые в свою очередь делятся на еще более мелкие шаги. Каждый шаг разбивается на две части, к первой части прибавляется сдвиг, посчитанный по начальной угловой частоте, вторая часть пересчитывается с учетом влияния коэффициентов других объектов. Таким образом получается довольно плавное движение.

После просчета всех данных, начинается создание графиков и схем. Для построения графиков используется библиотека matplotlib в python, а для создания анимации gif matplotlib.animation. Сначала описывается расположение графиков, их масштаб, положение названий и текстов, после этого в специальной функции на график покадрово наносятся

данные следующего времени. Эта функция используется сразу в команде для составления анимации gif.



Для того, чтобы программа подходила под условие задачи со спортсменами, было необходимо отредактировать значения угловой частоты (которая считается по линейной скорости спортсменов и радиусу поля $\omega = \frac{v}{R}$).



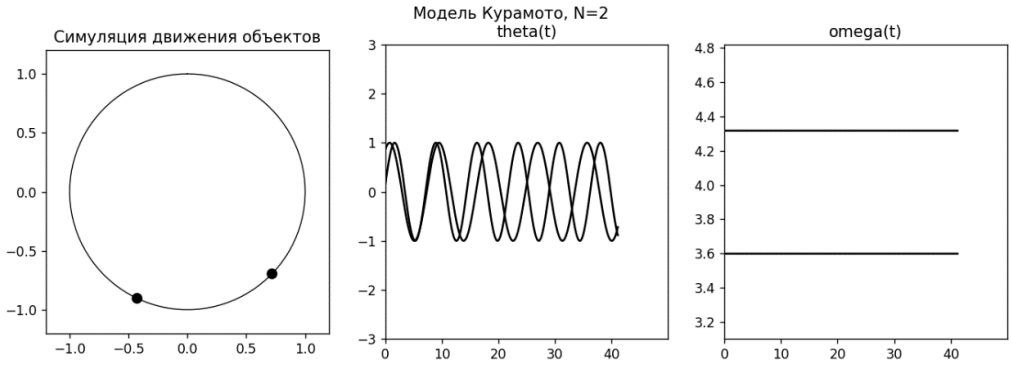
Результаты:

Проанализировав поведение модели Курамото при разных значениях связи K, можно сказать, что синхронизация при ненулевом коэффициенте может произойти не всегда. Стоит отметить, что синхронизация угловых частот не всегда сопровождается слиянием объектов в одну точку. При определенных условиях объекты могут находиться на большом расстоянии друг от друга, но при этом двигаться с одной скоростью. Это может быть связано с положением других объектов, их коэффициентами связи. Также в нашей задаче нужно учесть 2 фактора:

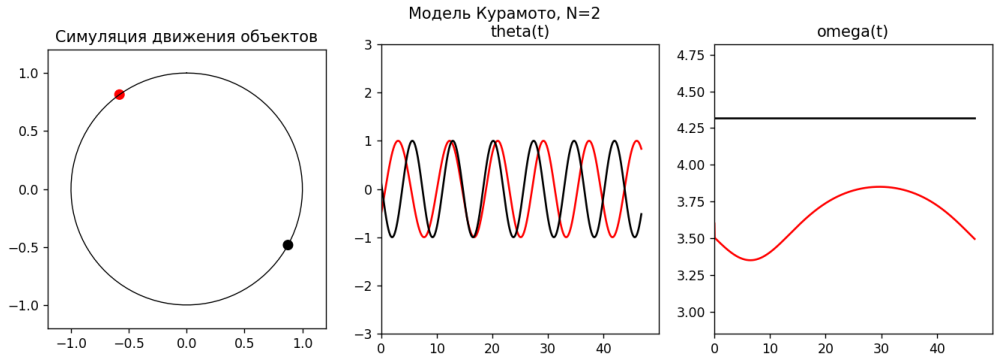
один из спортсменов равнодушен, то есть коэффициент связи у этого осциллятора равен нулю (пусть K = 0 у бегуна с большей скоростью);

на критические значения коэффициента связи влияет и значение радиуса R (причем, чем больше значение радиуса, тем критическое значение K меньше), поэтому примем его значение постоянным, равным 5 м.

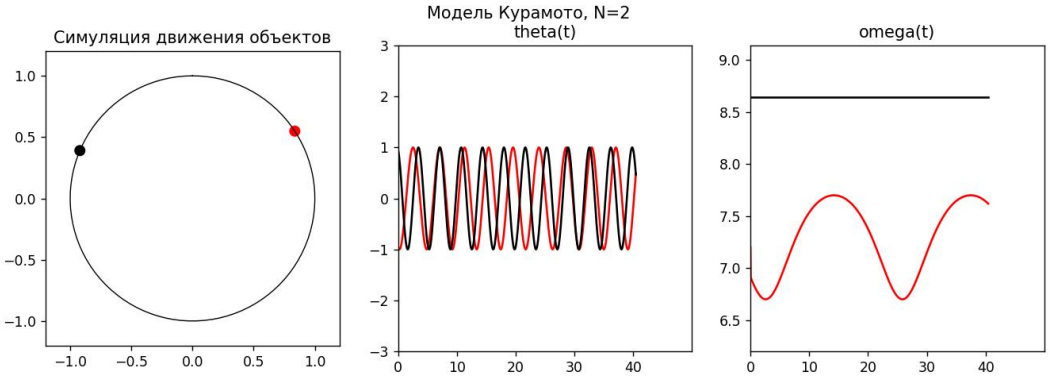
0) Очевидно, что при равенстве нулю коэффициента связи и второго осциллятора не будет происходить совершенно никакой синхронизации:



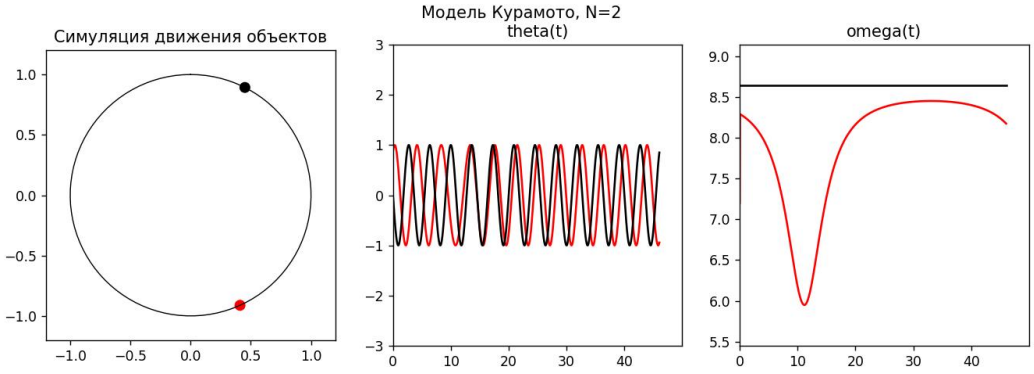
1) При K = 0.5 можно заметить «стремление» второго осциллятора к синхронизации, однако ее все равно не происходит:



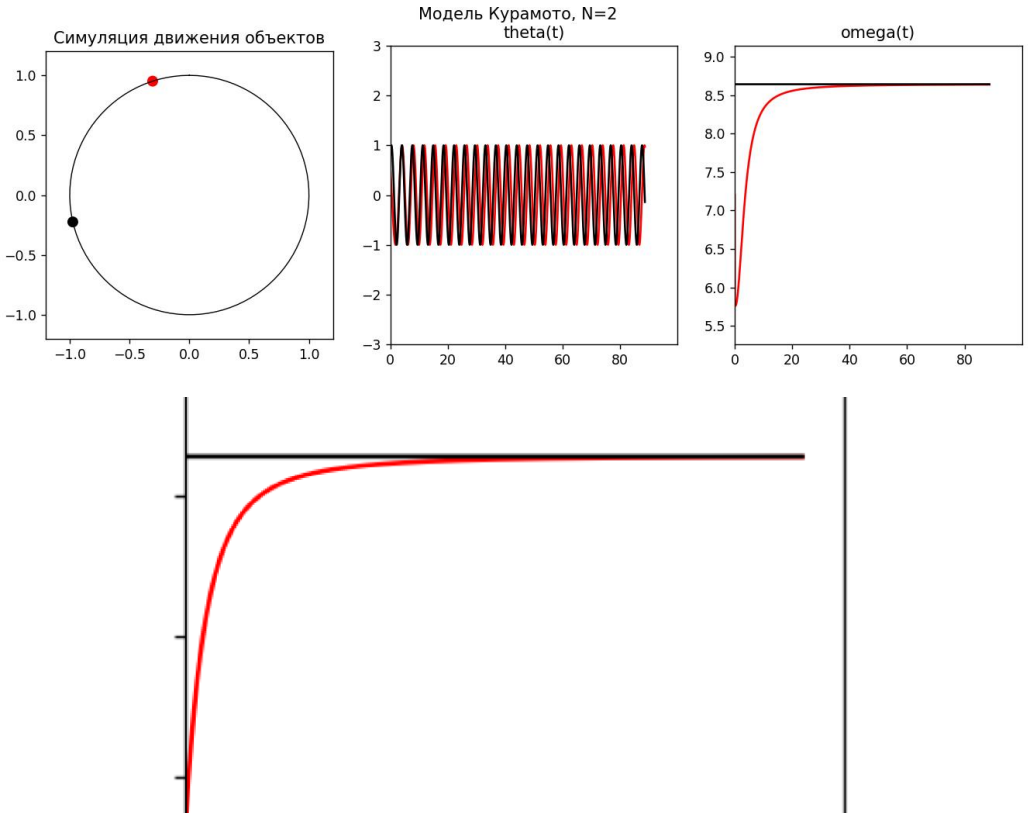
2) При K = 1 синхронизация также не происходит:



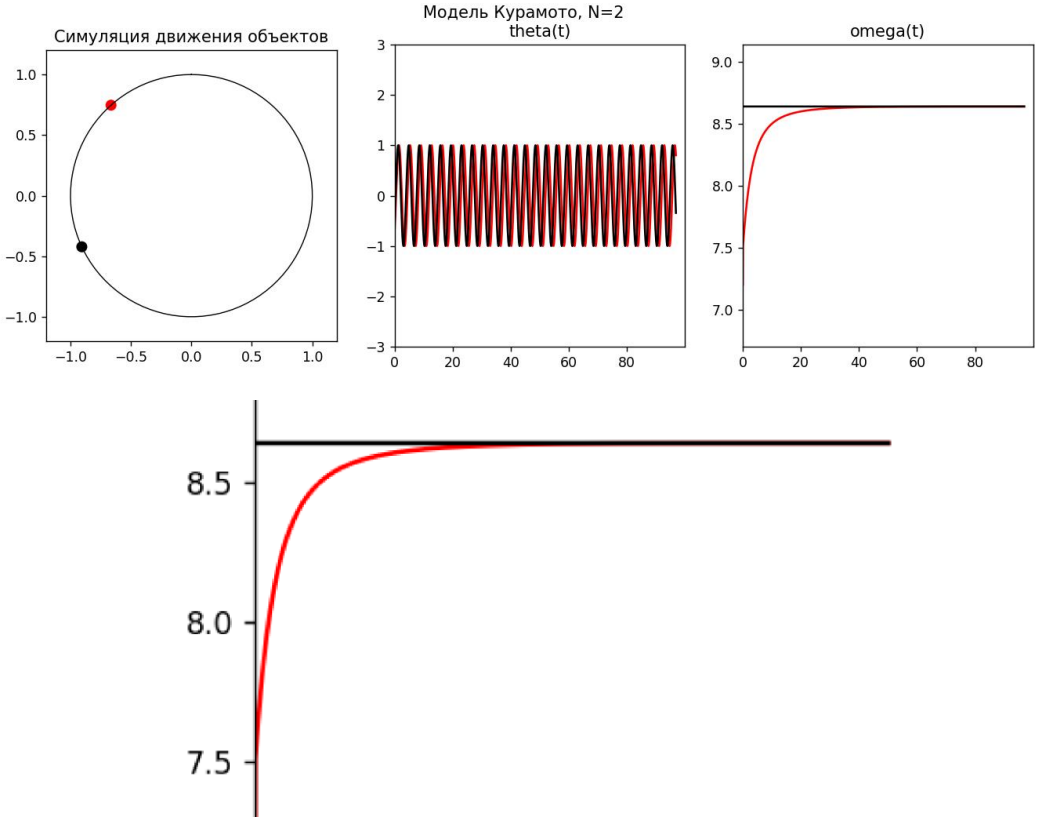
3) K = 2.5:



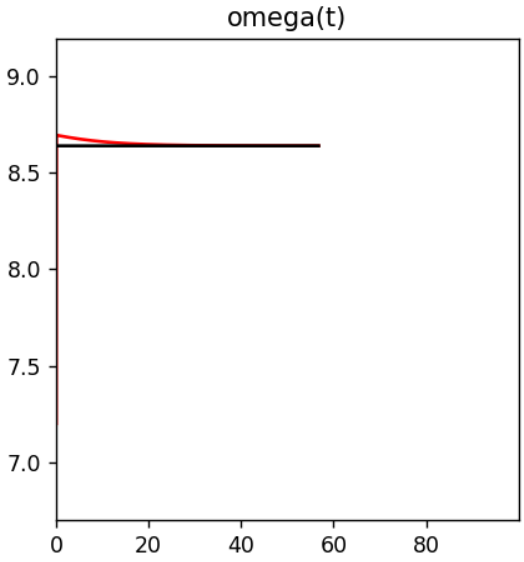
4) Взяв K = 2.9, может показаться, что синхронизация (по крайней мере по угловым частотам) происходит, однако это не так. Это можно увидеть даже на самой визуализации, масштабировав картинку:



5) K = 3 – это и есть наше критическое значение. При нем синхронизация происходит только в пределе, однако взяв любое значение немного больше 3, мы получим заветную синхронизацию по угловым частотам:

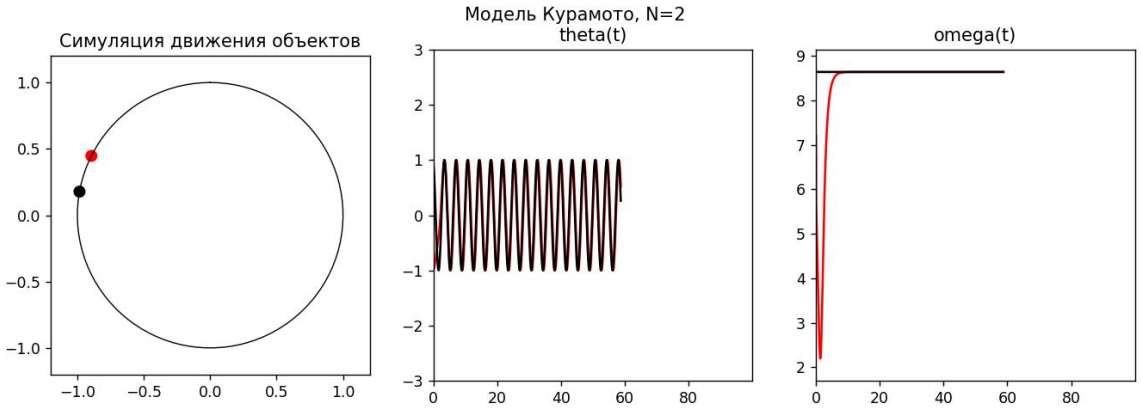


6) При К = 3.1 уже наблюдаем синхронизацию (на последней картинке это «идеально черная полоса»);



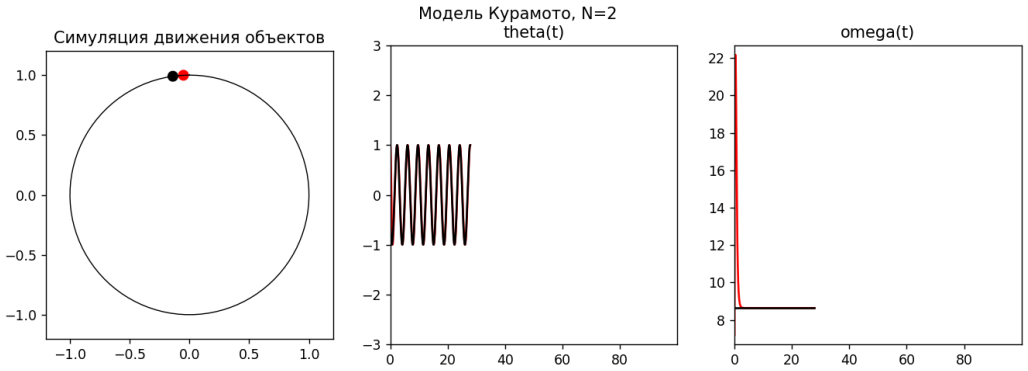
Далее, рассматривая значения К > 3, будем также получать синхронизацию, только быстрее. Также с увеличением значения коэффициента связи, объекты после синхронизацию начинают располагаться ближе к другу.

7) При К = 10:



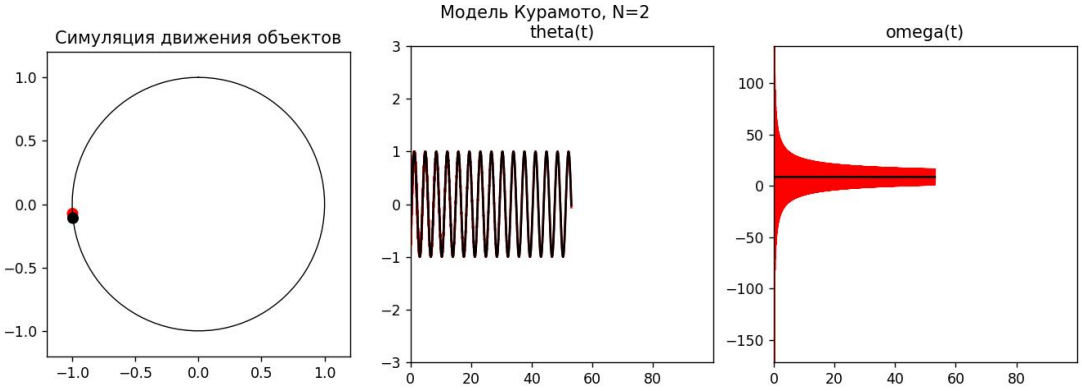
Интересно отыскать и такое значение приблизительное значение К, начиная с которого объекты будут двигаться как единое целое (фазы тоже станут равны).

8) Для нашей системы это критическое значение K = 30 (полная синхронизация по фазе видна из 2 картинки):



С учетом метода дифференцирования, можно отметить еще одну ключевую точку. Так как дифференцирование происходит путем деления времени на маленькие промежутки, то от того, насколько мелким выбран данный промежуток, будет зависеть плавность перемещения объектов и точность вычислений. Если мы возьмем очень большое K, то объекты начинают «перелетать» круг, и происходит рассинхронизация, которой в идеальных условиях вычислений, конечно же, не будет.

9) Начиная со значения K= 300 будет наблюдаться как раз таки этот эффект, однако не сразу заметен, поэтому для наглядности было взято K = 400:



Также задачей нашего проекта было выяснить, в каком случае синхронизация произойдет быстрее: если K = 0 будет у спортсмена с большей или меньшей скоростью. Для этого проведем 3 эксперимента, с одинаковыми входными параметрами K и значения радиуса. Так как нам бы хотелось узнать время начала полной синхронизации объектов (совпадение их фаз и угловых частот), то K берем достаточно большим.

Опыт 1: K = 200, R = 5

А) K = 0 у бегуна с меньшей скоростью:

```
D:\programs_py\Project_fiz\venv\Scripts\python.exe D:\programs_py\Project_fiz\main.py
Точное значение времени синхронизации: 929.14
```

Б) K = 0 у бегуна с большей скоростью:

```
D:\programs_py\Project_fiz\venv\Scripts\python.exe D:\programs_py\Project_fiz\main.py
Точное значение времени синхронизации: 356.89000000000004
```

Опыт 2: K = 250, R = 5

А) K = 0 у бегуна с меньшей скоростью:

```
D:\programs_py\Project_fiz\venv\Scripts\python.exe D:\programs_py\Project_fiz\main.py
Точное значение времени синхронизации: 729.09
```

Б) K = 0 у бегуна с большей скоростью:

```
D:\programs_py\Project_fiz\venv\Scripts\python.exe D:\programs_py\Project_fiz\main.py
Точное значение времени синхронизации: 190.44
```

Опыт 3: K = 300, R = 5

А) K = 0 у бегуна с меньшей скоростью:

```
D:\programs_py\Project_fiz\venv\Scripts\python.exe D:\programs_py\Project_fiz\main.py
Точное значение времени синхронизации: 593.1899999999999
```

Б) K = 0 у бегуна с большей скоростью:

```
D:\programs_py\Project_fiz\venv\Scripts\python.exe D:\programs_py\Project_fiz\main.py
Точное значение времени синхронизации: 79.69000000000001
```

В ходе эксперимента было выяснено, что синхронизация происходит быстрее при условии, что нулевым коэффициентом связи обладает бегун с большей скоростью

Вывод:

Мы смогли применить модель Курамото под описание динамики системы из связанных осцилляторов на примере спортсменов, бегущих по стадиону. К тому же удалось визуализировать такую модель на примере движущихся по кругу связанных между собой шариков, а также изменить ее под конкретную задачу. Были найдены критические точки K смены режимов, в том числе и те, которые связаны с недостатками выбранного метода численного дифференцирования.

Дополнительные файлы:

Файлы с кодом программы для общего случая и под конкретную задачу, а также анимации к ним, можно найти тут:

https://github.com/Vladislav909090/Model_Kuramoto-physics_project