Problem Set 2

Erik Norlin
Karl Lundgren

February, 2023

Problem 2.3

a) Here we expand the equation in the vicinity of the bifurcation in terms of r, then in terms of K. We then solve for C.

The branch starts to bifurcate from $r = 0$ and $K = K_c$. To obtain K_c we take the limit $r \to 0^+$.

$$\begin{aligned} 1 &= \lim_{r\to 0^+} K \int_{-\pi/2}^{\pi/2} d\theta \cos^2\theta g(Kr\sin\theta) \\ &= K_c \int_{-\pi/2}^{\pi/2} d\theta \cos^2\theta g(0) \end{aligned} \tag{1}$$

$$\Rightarrow K_c = \frac{1}{\int_{-\pi/2}^{\pi/2} d\theta \cos^2\theta g(0)} = \frac{1}{\frac{\pi}{2} g(0)} = \frac{1}{\frac{\pi}{2}\frac{1}{\pi\gamma}} = 2\gamma \tag{2}$$

To obtain r we expand eq. 1 around $K = K_c$ when $r = 0$ in powers of r and neglect higher order terms.

$$\begin{aligned} 1 &\approx K \int_{-\pi/2}^{\pi/2} d\theta \cos^2\theta \left(g(0) + g'(0)(Kr\sin\theta - 0) + \frac{1}{2} g''(0)(Kr\sin\theta - 0)^2 + O(...) \right) \\ &= K \int_{-\pi/2}^{\pi/2} d\theta \cos^2\theta \left(g(0) + 0 + \frac{1}{2} g''(0) K^2 r^2 \sin^2\theta \right) \\ &= \frac{K}{K_c} + K\frac{1}{2} g''(0) K^2 r^2 \int_{-\pi/2}^{\pi/2} d\theta \cos^2\theta \sin^2\theta \\ &= \frac{K}{K_c} + K\frac{1}{2}\frac{(-2)}{\pi\gamma^3} K^2 r^2 \frac{\pi}{8} \\ &= \frac{K}{K_c} - \frac{K^3 r^2}{8\gamma^3} \end{aligned} \tag{3}$$

$$\Rightarrow \frac{K^3 r^2}{8\gamma^3} \approx \frac{K}{K_c} - 1 \Rightarrow r^2 \approx \frac{8\gamma^3}{K^3}\left(\frac{K - K_c}{K_c}\right) \tag{4}$$

We expand K around K_c so $K = K_c + \eta$

$$\Rightarrow r^2 \approx \frac{8\gamma^3}{(K_c+\eta)^3}\left(\frac{(K_c+\eta)-K_c}{K_c}\right) = \frac{8\gamma^3\eta}{(K_c^3+3K_c^2\eta+3K_c\eta^2+\eta^3)K_c} \tag{5}$$

We neglect higher order terms of η

$$\Rightarrow r^2 \approx \frac{8\gamma^3\eta}{(K_c+3\eta)K_c^3} = \frac{8\gamma^3\eta}{(K_c+3\eta)(2\gamma)^3} = \frac{\eta}{K_c+3\eta} \tag{6}$$

We neglect the term 3η in the denominator because it barely contributes to K_c, while η in the numerator does.

$$\Rightarrow r^2 \approx \frac{\eta}{K_c} = \frac{K-K_c}{K_c} = \mu \Rightarrow r \approx \sqrt{\mu} \tag{7}$$

From the before we have that $r = C\sqrt{\mu}$ so therefore $C = 1$.

b) Simulations of the Kuramoto model

$$\dot{\theta}_i = \omega_i + \frac{K}{N}\sum_{j=1}^{N}\sin\left(\theta_j - \theta_i\right) \tag{8}$$

were carried out.

To sample points from the probability density distribution (PDF) we used the Inverse Transform Sampling Method as follows:

In this case the given probability density function $g(\gamma,\omega)$, was the Cauchy distribution. By integrating the PDF the cumulative density function, CDF, was obtained.

$$\int g(\gamma,\omega)\mathrm{d}\omega = \int \frac{\gamma}{\pi(\omega^2+\gamma^2)}\mathrm{d}\omega = \frac{\arctan\left(\frac{\omega}{\gamma}\right)}{\pi} \tag{9}$$

We then set

$$U[0,1] \equiv \frac{\arctan\left(\frac{\omega}{\gamma}\right)}{\pi} \tag{10}$$

where $U[0,1]$ is the uniform distribution sampled with values in the interval $[0,1]$ and invert the CDF $\Rightarrow$ CDF^{-1}

$$\omega_i = \gamma\tan(\pi U[0,1]). \tag{11}$$

We initialized N random points in the set $U[0,1]$ and then sampled from the inverted CDF. N random phases θ were initialized in the interval $[-\pi/2,\pi/2]$ sampled from the uniform distribution too.

The simulation was then carried out for T seconds. In each time step the Kuramoto Equation 8 was computed for all N oscillators. The derivatives, $\dot{\theta}_i$, were used to obtain the phases θ_i for the next time step using Euler's method for integration. The order parameter r was also computed in each time step. Since the phases exist in the real and

imaginary plane we get r by taking the norm of the average real components, $\cos\theta_n$, and imaginary components, $\sin\theta_n$, of all phases n, i.e.,

$$r = \frac{1}{N}\sum_{i=1}^{N}\|\mathrm{e}^{\mathrm{i}\theta_i}\| = \frac{1}{N}\sqrt{\left[\sum_{i=1}^{N}\cos(\theta_i)\right]^2 + \left[\sum_{i=1}^{N}\sin(\theta_i)\right]^2} \tag{12}$$

The simulation was carried out with all combinations for three numbers of oscillators $N \in \{20, 100, 300\}$ and for three values of the ratio $K/K_c \in \{0.5, 1.01, 1.5\}$, i.e., with a K below, close to and above the critical value K_c.

The results of the nine simulations can be seen in the figures below. In each simulation there are four quantities shown. Firstly, the simulated value of r vs. time t, secondly its average value ignoring a transient, thirdly the estimated value of $r = C\sqrt{\mu}$ derived in Section a) which will be accurate only close to K_c. Lastly, for the case where the Cauchy density function is used there is apparently an exact analytical expression for the order parameter which Kuramoto derived [**comp-bio**], simply

$$r = \sqrt{1 - \frac{K_c}{K}}. \tag{13}$$

This formula is valid in whole domain of K as long as we sampled from the Cauchy distribution $g(\omega)$ for the modes ω_i.

As can be seen by looking at the equation, it saturates towards unity as $K \gg K_c$, i.e., we get total synchrony as we move past K_c.

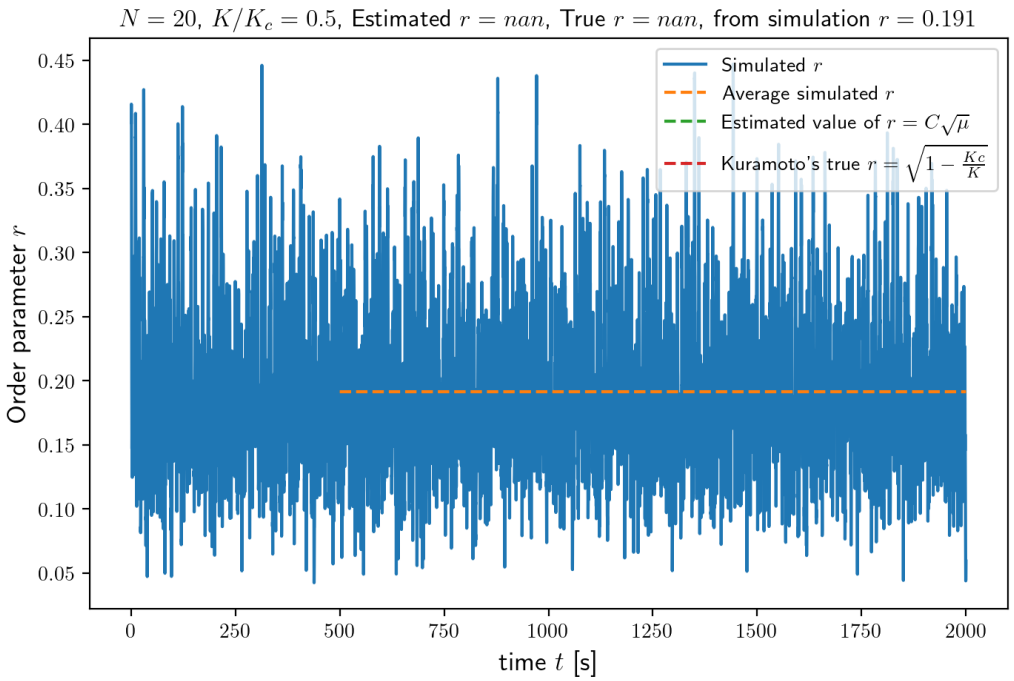

Figure 1: The evolution of the order parameter r over time t for the case where $N = 20$ and $K/K_c = 0.5$.

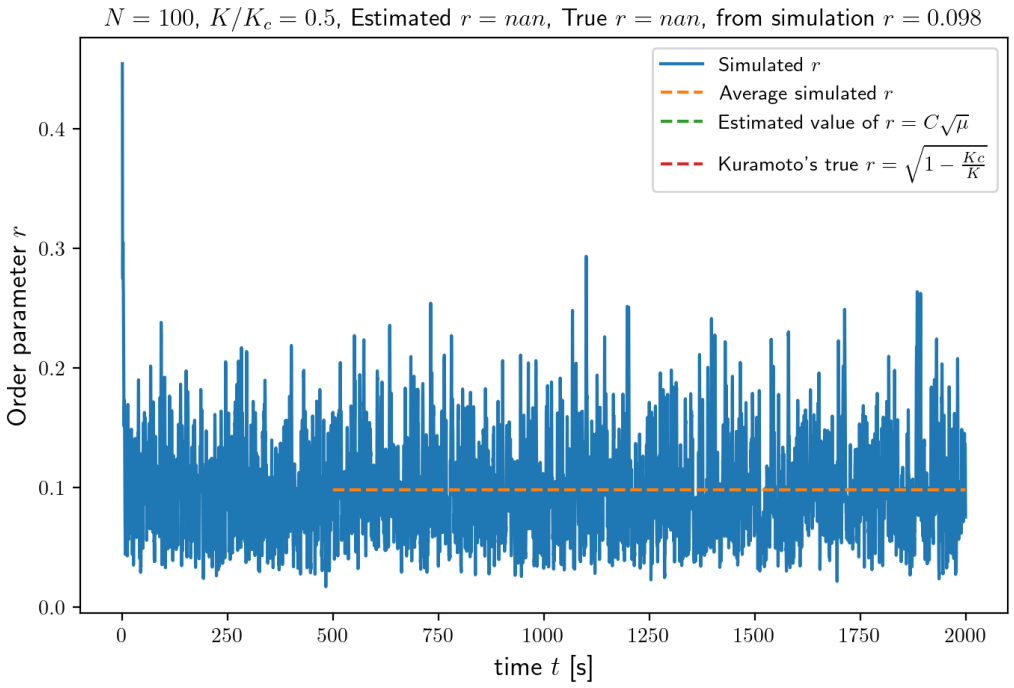

Figure 2: The evolution of the order parameter r over time t for the case where $N = 100$ and $K/K_c = 0.5$.

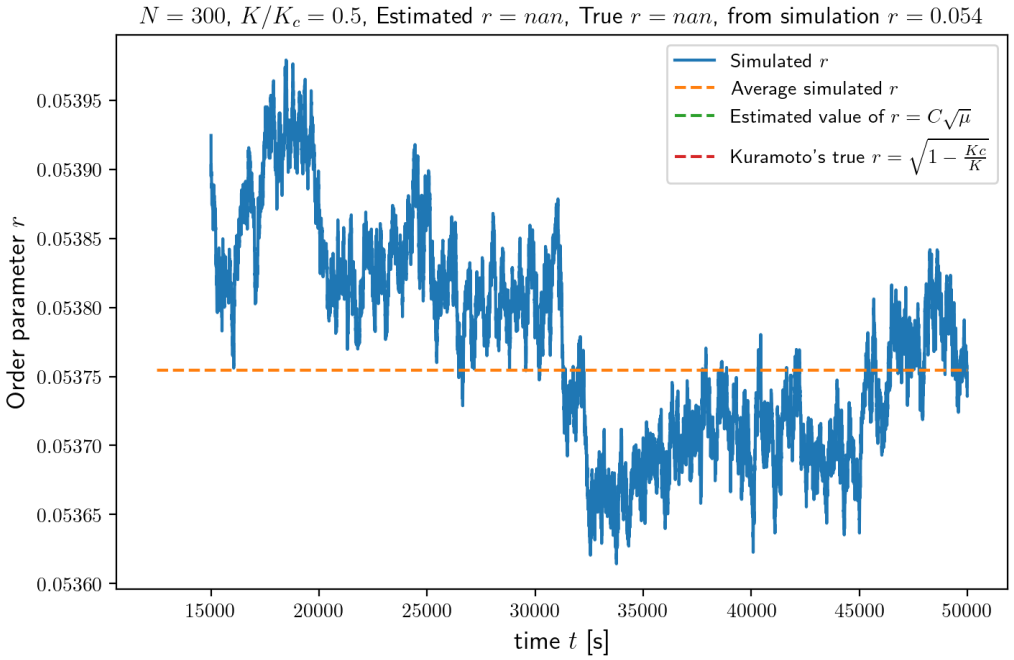

Figure 3: The evolution of the order parameter r over time t for the case where $N = 300$ and $K/K_c = 0.5$.

In the following Figures 1 to 3 when $K/K_c = 0.5$ it can be seen that r is a small value, although still quite far from zero. In theory, when $K/K_c \leq 1$, the order of the system should tend towards zero, more formally, $r_\infty = 0$. Also as N increases we get lower values of r, i.e., for higher resolution the result of the simulation estimates the theory better.

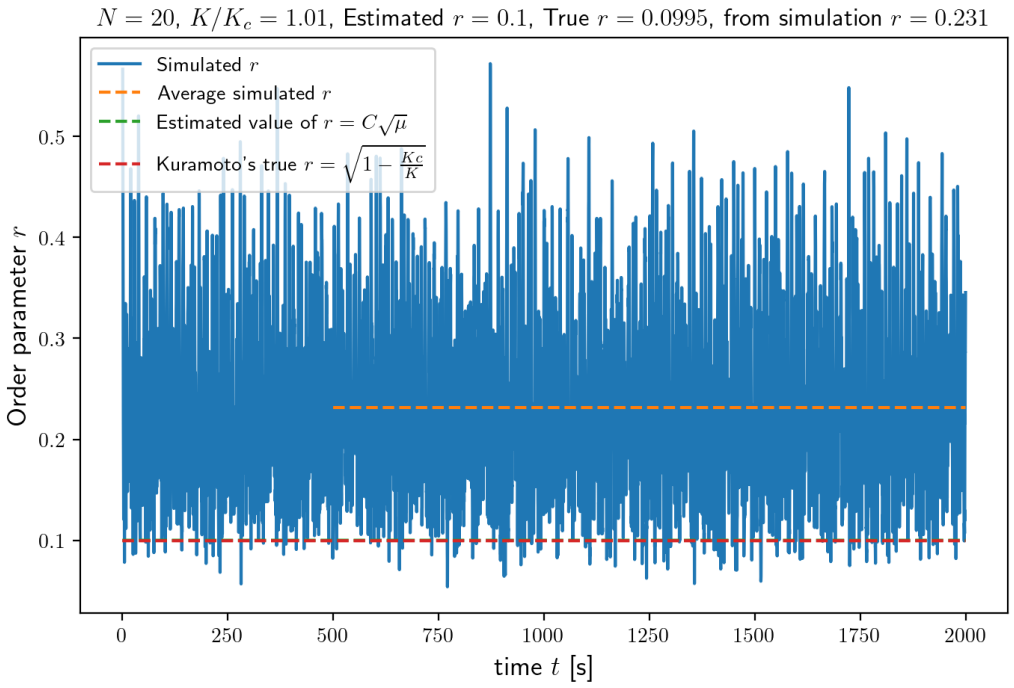

Figure 4: The evolution of the order parameter r over time t for the case where $N = 20$ and $K/K_c = 1.01$.

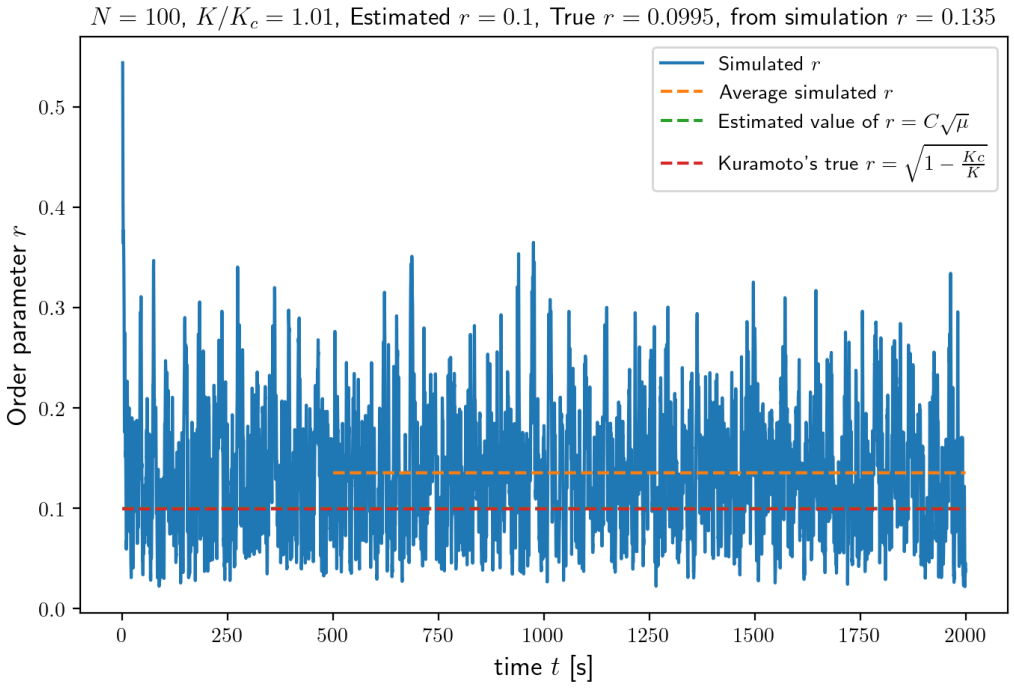

Figure 5: The evolution of the order parameter r over time t for the case where $N = 100$ and $K/K_c = 1.01$.

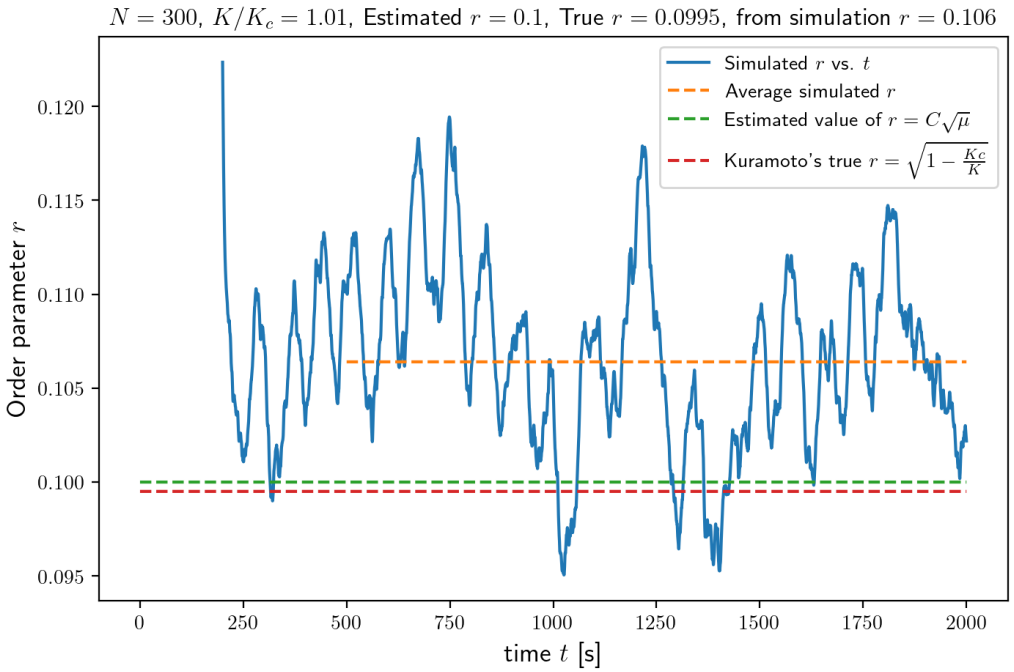

Figure 6: The evolution of the order parameter r over time t for the case where $N = 300$ and $K/K_c = 1.01$.

Continuing for the case where $K \approx K_c$, the value of r should be very small. Analytically, both the approximation $r = C\sqrt{\mu}$ and Kuramoto's formula (13) should be consistent with each other here. Again, as can be seen from the Figures 4 to 6 as N increases the simulation approximates the analytical value for r while the fluctuations also decrease.

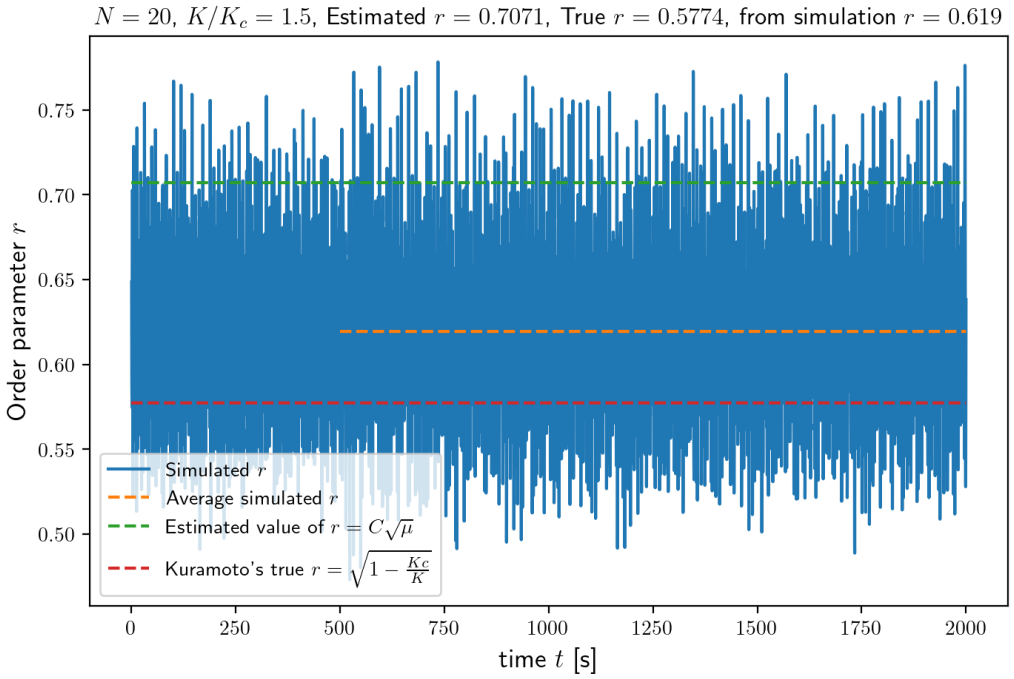

Figure 7: The evolution of the order parameter r over time t for the case where $N = 20$ and $K/K_c = 1.5$.

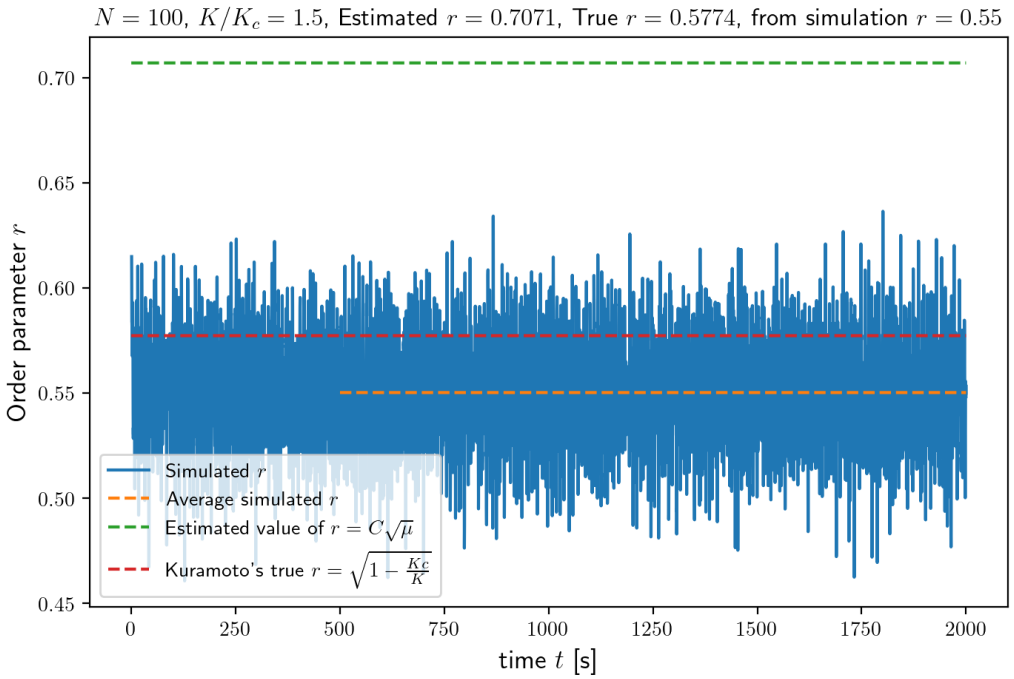

Figure 8: The evolution of the order parameter r over time t for the case where $N = 100$ and $K/K_c = 1.5$.

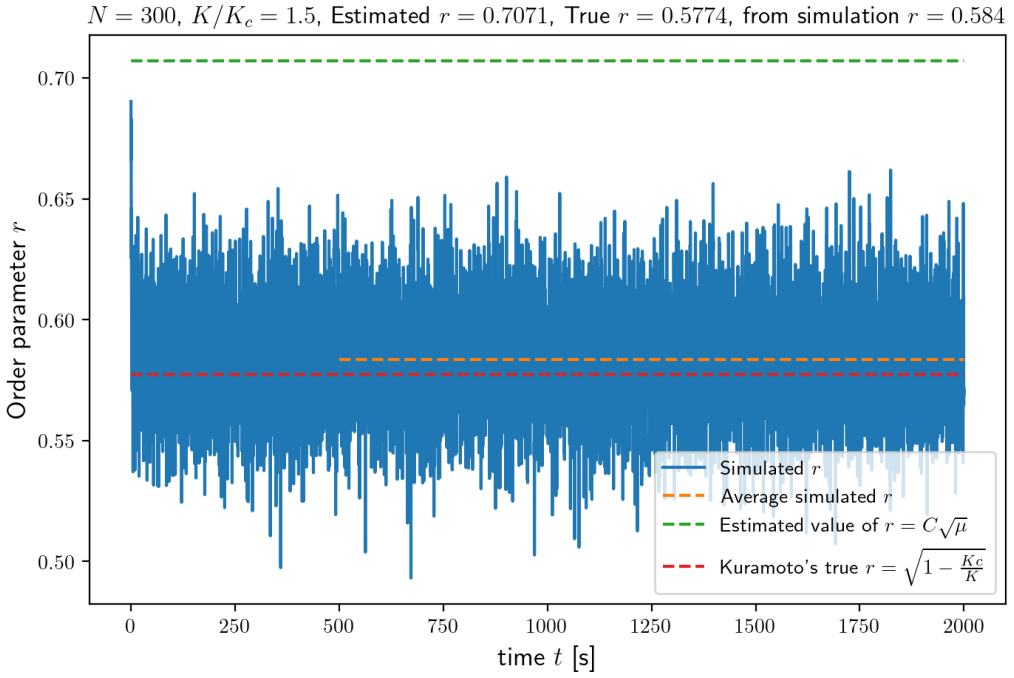

Figure 9: The evolution of the order parameter r over time t for the case where $N = 300$ and $K/K_c = 1.5$.

Lastly, for the case $K/K_c = 1.5$, we are now quite far away from the critical value K_c. The approximation $r = C\sqrt{\mu}$ should fail here while Kuramoto's formula (13) should still be valid. From the figures 7 to 9 it can be seen that the numerical simulation confirms this. Yet again, as number of oscillators N increase the simulation approximates Kuramoto's analytical value for r

while the fluctuations also decrease.

Appendix

Python code for simulations of section b)

```
import numpy as np
import matplotlib.pyplot as plt
from tqdm.notebook import tqdm
%matplotlib inline
%config InlineBackend.figure_format = 'retina'
plt.rcParams['text.usetex'] = True

T = 2000
dt = 0.2
N = 20

gamma = 1
Kc = 2 * gamma
multiplier = 1.01
K = multiplier * Kc
mu = (K - Kc) / Kc
phase_list = np.random.uniform(-np.pi / 2, np.pi / 2, N)
U = np.random.uniform(0, 1, N)
omega_list = gamma * np.tan(np.pi * U)
dtheta_dt = np.zeros_like(omega_list)

t_list = np.linspace(0, T, int(T / dt))
r_list = np.zeros(len(t_list))

# r when t=0
real_part = np.sum(np.cos(phase_list))
imaginary_part = np.sum(np.sin(phase_list))

for t in tqdm(range(len(t_list))):
    for i in range(N):
        omega = omega_list[i]
        phase = phase_list[i]
        dtheta_dt[i] = omega + (K / N) * np.sum(np.sin(phase_list - phase))

    phase_list = phase_list + dtheta_dt * dt
    real_part = np.sum(np.cos(phase_list)) / N
    imaginary_part = np.sum(np.sin(phase_list)) / N
    r_list[t] = np.sqrt(real_part ** 2 + imaginary_part ** 2)

import pandas as pd

fig, ax = plt.subplots(figsize=(8, 5))

rDF = pd.DataFrame(r_list)
window = int(T * dt * N / 1000)
window = 5
```

```
rDF_moving_average = rDF.rolling(window=window).mean()

# ax.plot(t_list, r_list, label=r'Simulated $r$ vs. $t$')
ax.plot(t_list, rDF_moving_average, label=r"Simulated $r$")
ax.set_xlabel("time $t$ [s]", fontsize=13)
ax.set_ylabel("Order parameter $r$", fontsize=13)

simulatedAverage = np.mean(r_list[int(len(t_list) * 0.25) :])

ax.plot(
    t_list[int(len(t_list) * 0.25) :],
    simulatedAverage * np.ones_like(t_list[int(len(t_list) * 0.25) :]),
    linestyle="--",
    label=r"Average simulated $r$",
)

r_est = np.sqrt(K / Kc - 1)
ax.plot(
    t_list,
    r_est * np.ones_like(t_list),
    linestyle="--",
    label=r"Estimated value of $r = C \sqrt{\mu}$",
)

r_true = np.sqrt(1 - Kc / K)
ax.plot(
    t_list,
    r_true * np.ones_like(t_list),
    linestyle="--",
    label=r"Kuramoto's true $r = \sqrt{1 - \frac{Kc}{K}}$",
)

ax.set_title(
    r"$N={}$, $K/K_c={}$, Estimated $r={}$, True $r = {}$, from simulation $r
        ↪ ={}$".format(
        N, multiplier, round(r_est, 4), round(r_true, 4), round(
            ↪ simulatedAverage, 3)
    )
)
plt.legend()


plt.show()
# mul = np.linspace(1,2)
# K = mul * Kc

# r_analytic = np.sqrt(Kc**2 * (K - Kc) / K**3)
# print(r_analytic)
# r_cauchy = np.sqrt(1 - Kc / K)
# r_cauchy
```

```
# plt.plot(K, r_cauchy)
# plt.plot(K, r_analytic)

K = multiplier * Kc
```