The Kuramoto Model:
A Non-Linear Dynamics Exercise*

Monit Sharma
Indian Institute of Science Education And Research
Mohali
(Dated: March 29, 2021)

The Kuramoto model depicts a large population of the coupled oscillators with some natural frequencies in a specific distribution.

Whenever the coupling strength passes the threshold value, the entire oscillatory system observes a phase transition. The oscillators synchronize with one another, while few stay incoherent. This model has its use in mathematical biology, statistical physics, plasma physics and much more.

I. INTRODUCTION

The Kuramoto model was proposed by Yoshiki Kuramoto and is used to explain the synchronization phenomena. It explicitly defines the behaviour of coupled oscillators, and that's why it's used for describing chemical and biological oscillators and has wide use in neuroscience. In Physics, it's used to explain the Josephson Junctions.

Although there are several assumptions made for the model, like weak coupling and identical oscillators.

II. BACKGROUND

The Kuramoto model was initially made for explaining the collective synchronization behaviour, simply the phenomenon when the extensive system of oscillators locks themselves to a standard frequency spontaneously, given the difference in their natural frequencies. This is used in various fields. In Biology, we have pacemakers for the heart, circadian pacemakers for the brain, physics it's wide range include LASERs, microwave oscillations and superconducting Josephson Junctions.

The phenomenon of collective synchronization was first studied by Wiener, who understood its true potential and general area of use. He used the technique of Fourier integrals for describing collective synchronization, but it wasn't much fruitful.

Winfree made a more fruitful approach. He formulated the problem as interacting limit-cycle oscillators, and he made assumptions like weak coupling and nearly identical oscillators that helped simplify the problem. Winfree supposed that the individual oscillator was coupled to the collective rhythm of the entire system.

The Winfree model is:

$$\dot{\theta}_i = \omega_i + \left(\sum_{j=1}^{N} X(\theta_j) \right) Z(\theta_i),$$

Here $\dot{\theta}_i$ denotes the phase of the oscillator i and ω_i its natural frequency. Each oscillator j exerts a phase-dependent influence $X(\theta_j)$ on all the others: and the response of the oscillator i depends on its phase θ_i through the sensitivity function $Z(\theta_i)$.

With numerical simulations and approximations, Winfree found that the oscillator populations could show the temporal analog of Phase transition. Whenever the natural frequency spread is large compared to coupling, the system shows incoherent behaviour, where each oscillator is running on its natural frequency. When the spread decrease, the incoherence remains till it crosses the threshold value. After the threshold value, the smaller cluster of oscillation freezes and gets synchronous.

This whole phenomenon made Kuramoto write a paper on it.

III. KURAMOTO MODEL

A. Mathematical Equations

Kuramoto used Winfree's intuition about the model and made a firmer foundation. He used the perturbative method of averaging and the assumptions of weak coupling and identical oscillators and showed that the long term dynamics are given by:

$$\dot{\theta}_i = \omega_i + \left(\sum_{j=1}^{N} \Gamma_{ij}(\theta_j - \theta_i) \right)$$

Where the interaction functions Γ_{ij} can be calculated as integrals containing terms from original model.

Even after all the assumptions and simplifications, this equation was still challenging to analyse. Since the interaction term contains an arbitrary number of Fourier harmonics and the connection topology was unspecified, the oscillators can be on a ring, a cubic lattice, or an unknown graph.

Kuramoto understood the mean-field case as the most moldable. The Kuramoto model corresponds to the sim-

*

plest case of pure sinusoidal coupling:

$$\Gamma_{ij}(\theta_j - \theta_i) = \frac{K}{N} sin(\theta_j - \theta_i),$$

where $K > 0$ refers to the coupling strength and the $1/N$ factor ensured that the model behaves well in the $N \to \infty$ limit.

The frequencies ω_i are distributed according to the probability density function $g(\omega)$. Kuramoto assumed that the probability density function is unimodal and symmetric around its mean frequency Ω, i.e $g(\Omega + \omega) = g(\Omega - \omega)$ for all ω, exactly like the gaussian distribution. And with the help of rotational symmetry in our model, we can set the mean frequency $\Omega = 0$, by setting $\theta_i \to \theta_i + \Omega t$, which means a rotating frame at frequency Ω. Now the Equation becomes invariant:

$$\dot{\theta}_i = \omega_i \frac{K}{N} sin(\theta_j - \theta_i),$$

but subtracts Ω from all the ω_i and thus the mean shifts from $g(\omega)$ to zero. So we get,

$$g(\omega) = g(-\omega)$$

for all the ω, and the ω_i along the deviation from the mean frequency of Ω. Another assumption made in the model is that $g(\omega)$ is no where increasing on $[0, \infty)$

B. Order Parameters

To visualize the phases' dynamics, it is convenient to imagine a pack of points running around the complex plane's unit circle. The complex order parameter is a small quantity that can be described as the collective rhythm produced by the whole population. The radius $r(t)$ measures the coherence of the phase and $\psi(t)$ is the average phase.

$$re^{i\psi} = \frac{1}{N} \sum_{j=1}^{N} e^{i\theta_j}$$

For example, if every oscillator drive in a particular tight bunch, we have $r = 1$, and the group acts as a large oscillator. Contrastingly, when the oscillators are dispersed around the circle, then $r = 0$; the particular oscillations add incoherently, and no visual rhythm is presented. Kuramoto noticed that the governing equation.

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

can be rewritten in the order parameter terms, by equating the Imaginary part of both

$$rsin(\psi - \theta_i) = \frac{1}{N} \sum_{j=1}^{N} sin(\theta_j - \theta_i)$$

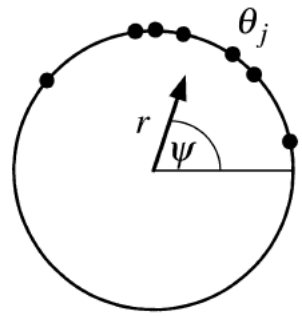


FIG. 1. Geometric Interpretation of Order Parameter

Thus, the equation becomes

$$\dot{\theta}_i = \omega_i + Krsin(\psi - \theta_i)$$

In this form, the mean-field part of the model becomes apparent. Each oscillator seems to be uncoupled. Although they are interacting, only through the mean-field quantities r and ψ. Precisely, the phase θ_i is pulled to the mean phase ψ rather than near the stage of any particular oscillator. Moreover, the active power of coupling is proportional to the coherence. This proportionality places up a connection between coupling and coherence; as the population grows more coherent, r grows. The effective coupling Kr increases, which leads to recruit even more oscillators the synchronized pack. If the recruits further increase the coherence, the process will continue; otherwise, it becomes self-limiting.

C. Simulations

If we integrate the model numerically, we will learn how $r(t)$ evolves. Suppose we fix $g(\omega)$ to be a Gaussian with infinite tails and then vary the coupling K. Simulations tell us that for all K more minor than a certain threshold K_c, the oscillators behave as if they are uncoupled: the phases get uniformly distributed around the circle, starting from any initial condition. Then the $r(t)$ value decays to a slight jitter.

But when the K exceeds K_c value, this incoherent state becomes unstable, and $r(t)$ grows exponentially, following the nucleation of a small cluster of oscillators that are mutually synchronized, thereby generating a collective oscillation. Eventually, $r(t)$ saturates at some level fluctuations.

At the level of the unique oscillators, we find that the population divides into two groups: the oscillators near the centre of the frequency distribution lock together at the mean frequency and co-rotate with the intermediate phase (t), while those in the tails run near their natural frequencies and drift relative to the synchronized cluster. This mixed state is often called partially synchronized. With further increases in K, more and more oscillators are recruited into the synchronized cluster.

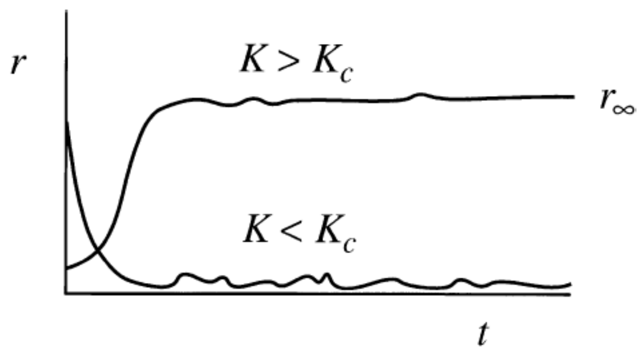


FIG. 2. Schematic Illustration

IV. KURAMOTO'S ANALYSIS

Kuramoto examined his model without the advantage of simulations. He chose the correct long-term behaviour of the limit $N \to \infty$, using symmetry concerns.

He attempted steady solutions, where $r(t)$ is constant and $\psi(t)$ rotates uniformly at frequency Ω. Using the rotating frame with frequency Ω and choosing the origin of this frame correctly, one can set $\psi = 0$ without loss of generality.

Then the governing equation becomes

$$\dot{\theta}_i = \omega_i - Krsin(\theta_i)$$

Since r is assumed constant, all the oscillators are effectively independent. The plan now is to work for the resulting motions of all the oscillators. These motions, in turn, mean values for r and which must be consistent with the values initially assumed. This self-consistency condition is the solution to the analysis.
The solution shows two type of the long term behaviour, with $|\omega_i| < Kr$ approaching a fixed point is defined

$$\omega_i = Krsin(\theta_i)$$

where $|\theta_i| < \frac{1}{2}\pi$.

All these oscillators will be locked because they all are in the same phase, or we can say they all are phase-locked at Ω . Contrastingly, the oscillators with the $|\omega_i| > Kr$ are all running around in circles in a non-uniform manner. The locked oscillators correspond to the centre of $g(\omega)$, and the drifting oscillators correspond to the tails.

V. UNSOLVED PROBLEMS

A. Synchronization in the $N \to \infty$ limit

Kuramoto's case included some spontaneous leaps that were distant from obvious, they began to appear obscure the more one thought about them.

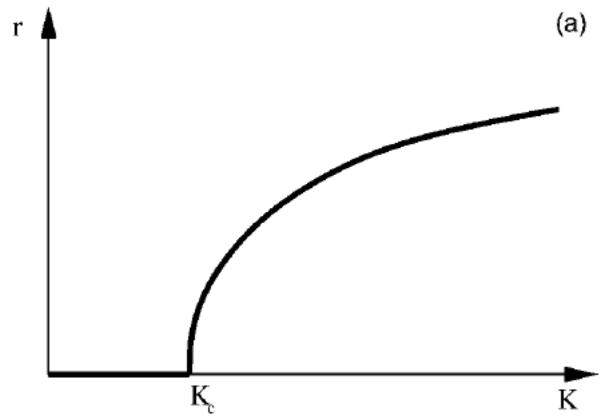


FIG. 3.

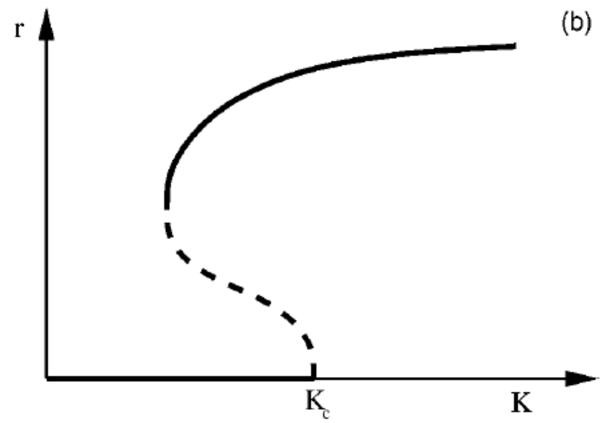


FIG. 4.

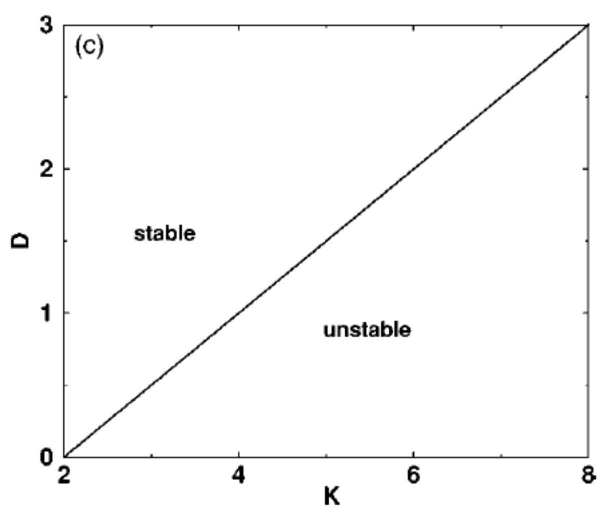


FIG. 5. Bifurcation and stability of the Kuramoto model:(a) supercritical bifurcation in a diagram of r vs K; (b) subcritical bifurcation; (c) phase diagram of the noisy Kuramoto model, D vs K,

Kuramoto's approach was based on the fact that r was a constant, but it cannot be strictly true for any finite N .

For a simple case of $K = 0$, we have $\dot{\theta}_i = \omega_i$, and every trajectory is dense. Accordingly $r(t)$ passes through every possible value between 0 and 1, which is contradicting to Kuromata's assumption of r to be a constant. $r(t)$ would be mostly found near 0 and only blip up sometimes.

The issue of fluctuation is still open Mathematically.

B. Stability

The other significant issue left unsolved by Kuramoto's analysis involves the stability of the steady answers. Kuramoto was aware of the stability problem. One may expect that weaker coupling makes the zero solution stable and stronger coupling unstable. Surprisingly enough, this seemingly obvious fact seems challenging to prove. The tricky part here is that an infinitely large number of phase configurations belong to an identical "macroscopic" state specified by a given value of r.

VI. KURAMOTO–SIVASHINSKY EQUATION

The Kuramoto–Sivashinsky equation (also called the flame equation) is a fourth-order nonlinear partial differential equation. It is titled after Yoshiki Kuramoto and Gregory Sivashinsky, who derived the equation from modelling the diffusive instabilities in a laminar flame front. The Kuramoto–Sivashinsky equation is known for its chaotic behaviour.

A. Maths

For 1-D version of the KS Equation:

$$u_t + u_{xx} + u_{xxx} + \frac{1}{2}u_x^2 = 0$$

An alternate version for the same equation is obtained by differentiating with respect to x and substituting $v = u_x$, mostly used in Fluid Dynamics.

$$v_t + v_{xx} + v_{xxxx} + vv_x = 0$$

This equation can be generalized to higher dimensions, by:

$$u_t + \Delta u + \Delta^2 u + \frac{1}{2}|\Delta u|^2 = 0$$

where Δ is the Laplace operator and Δ^2 is the biharmonic operator.

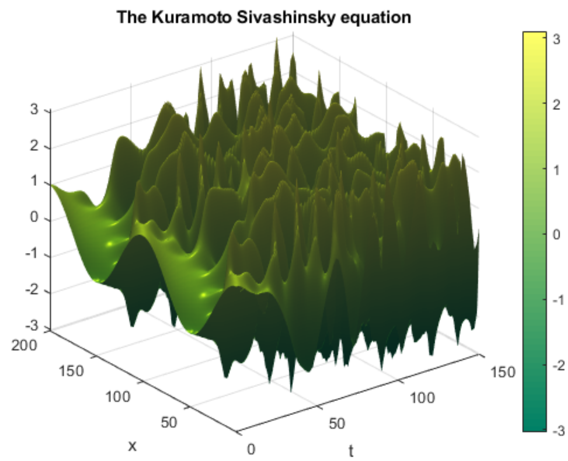


FIG. 6. The Kuramoto-Sivashinsky Equation

B. Solutions

Solutions of the Kuramoto–Sivashinsky equation hold rich dynamical properties.

They are considered on a periodic domain $0 < x < L$, the dynamics undergo a series of bifurcations as the domain size L is increased, culminating in the onset of chaotic behaviour. Depending on the value of L, solutions may include equilibria, relative equilibria, and travelling waves—all of which typically become dynamically unstable as L is increased. In particular, the transition to chaos occurs by a cascade of period-doubling bifurcations.

C. Applications

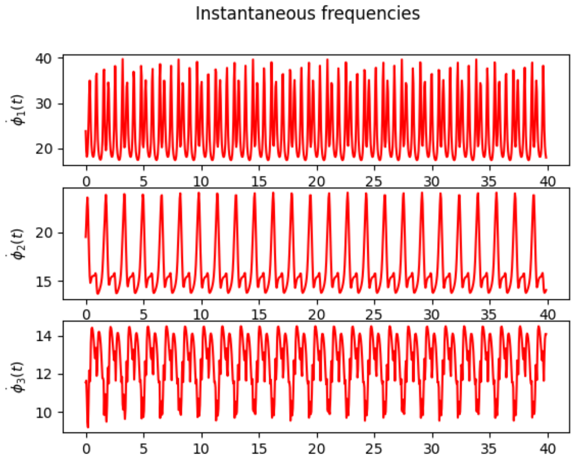
Applicability of the Kuramoto–Sivashinsky equation spread exceeding its unique context of flame propagation and reaction-diffusion systems. These new applications introduce flows in pipes and at interfaces, plasmas, chemical reaction dynamics, and models of ion-sputtered surfaces.

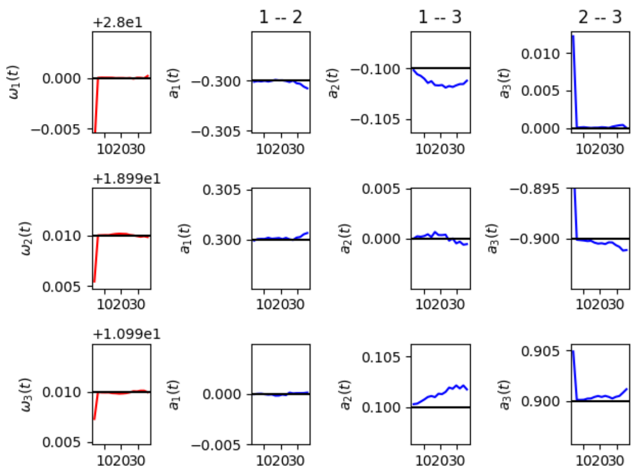
VII. BAYESIAN INTERFERENCE IN KURAMOTO MODEL

Technique refers to general phase dynamics in the oscillator, i.e.

$$\dot{\phi}_i = \omega_i + f_i(\phi_i) + g_i(\phi_i, \phi_j) + \xi_i$$

where $\omega_i, f_i(\phi_i)$ and $g_i(\phi_i, \phi_j)$ are intrinsic frequency, self-coupling and coupling with other oscillators, respectively. Term ξ_i refers to noise.



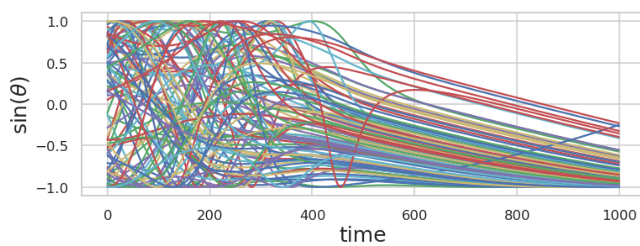


A. Procedure

- Rewrite all the equations of the model in the form $\dot{\phi}_i = \sum_{k=-K}^{K} C^{(}_{k}i) P_{i,k}(\Phi) + \xi_i(t)$, where C is the Parameter vector , and P provides significant terms for the model and Φ is a vector of all oscillators' phase.
- Calculate the diagonal values of Jacobian matrix, i.e $\frac{dP_{i,k}}{d\phi_i}$
- Set probabilities for parameters C vector and its covariance $\sum$

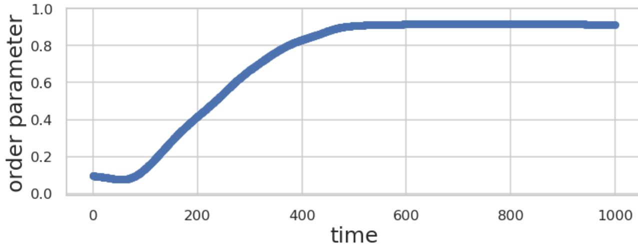
The problem is reduced to finding maximum minus log-likelihood function S.

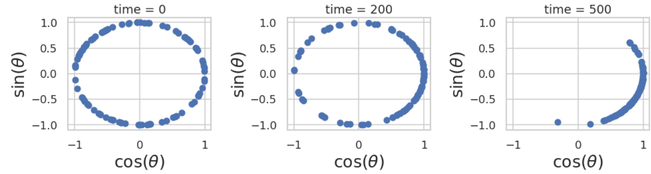
```
import numpy as np
import matplotlib.pyplot as plt
import networkx as nx
import seaborn as sns

from kuramoto import Kuramoto

plt.style.use('seaborn')
sns.set_style("whitegrid")
sns.set_context("talk")


# Instantiate a random graph and transform into an adjacency matrix
graph_nx = nx.erdos_renyi_graph(n=100, p=1) # p=1 -> all-to-all connectivity
graph = nx.to_numpy_array(graph_nx)

# Instantiate model with parameters
model = Kuramoto(coupling=3, dt=0.01, T=10, n_nodes=len(graph))

# Run simulation - output is time series for all nodes (node vs time)
act_mat = model.run(adj_mat=graph)

# Plot all the time series
plt.figure(figsize=(12, 4))
plt.plot(np.sin(act_mat.T))
plt.xlabel('time', fontsize=25)
plt.ylabel(r'$\sin(\theta)$', fontsize=25)
```



VIII. CODE AND IMPLEMENTATION

```
# Plot evolution of global order parameter R_t
plt.figure(figsize=(12, 4))
plt.plot(
    [model.phase_coherence(vec)
     for vec in act_mat.T],
    'o'
)
plt.ylabel('order parameter', fontsize=25)
plt.xlabel('time', fontsize=25)
plt.ylim((-0.01, 1))
```

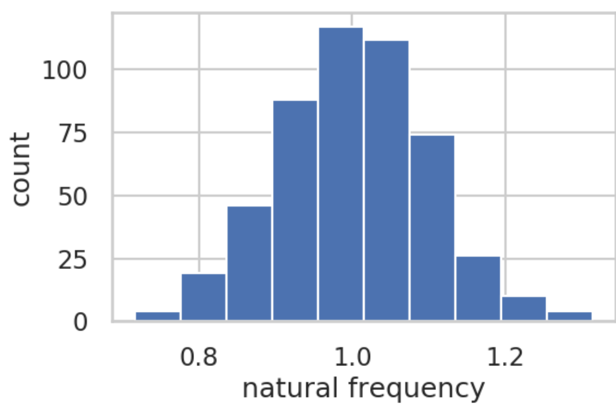




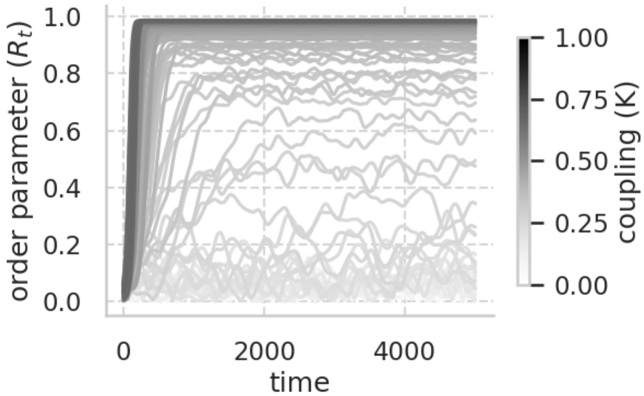

```python
# Plot oscillators in complex plane at times t = 0, 250, 500
fig, axes = plt.subplots(ncols=3, nrows=1, figsize=(14, 4))

times = [0, 200, 500]
for ax, time in zip(axes, times):
    ax.plot(np.cos(act_mat[:, time]),
            np.sin(act_mat[:, time]),
            'o', markersize=10)
    ax.set_title(f'time = {time}')
    ax.set_ylim((-1.1, 1.1))
    ax.set_xlim((-1.1, 1.1))
    ax.set_xlabel(r'$\cos(\theta)$', fontsize=25)
    ax.set_ylabel(r'$\sin(\theta)$', fontsize=25)
    ax.grid(True)
plt.tight_layout()
```


[1] Check the code on my Github.
[2] J.D. Crawford, J. Statist. Phys. 74 (1994) 1047.
[3] Y. Kuramoto, in: H. Arakai (Ed.), International Symposium on Mathematical Problems in Theoretical Physics, Lecture Notes in Physics, Vol. 39, Springer, New York, 1975, p. 420.
[4] Y. Kuramoto, I. Nishikawa, in: H. Takayama (Ed.), Cooperative Dynamics in Complex Physical Systems, Springer, Berlin, 1989
[5] C.S. Peskin, Mathematical Aspects of Heart Physiology, Courant Institute of Mathematical Science Publication, New York, 1975, pp. 268–278.
[6] S.Yu. Kourtchatov, V.V. Likhanskii, A.P. Napartovich, F.T. Arecchi, A. Lapucci, Phys. Rev. A 52 (1995) 4089.
[7] J.D. Crawford, in: W. Greenberg, J. Polewezak (Eds.), Modern Mathematical Methods in Transport Theory, Operator Theory: Advances and Applications, Vol. 51, Birkhauser, Basel, 1991, p. 97.
[8]
[8] K. Wiesenfeld, P. Colet, S.H. Strogatz, Phys. Rev. E 57 (1998) 1563.
[9] From Kuramoto to Crawford: exploring the onset of synchronization in populations of coupled oscillators Steven H. Strogatz
[10] Z. Jiang, M. McCall, J. Opt. Soc. Am. 10 (1993) 155.
[11] Kuramoto, Y. (1984). Chemical Oscillations, Waves, and Turbulence (Vol. 19). http://doi.org/10.1007/978-3-642-69689-3
[12] Acebron, J. A. et al (2005). The Kuramoto model: A simple paradigm for synchronization phenomena. Reviews of Modern Physics, 77(1), 137–185. http://doi.org/10.1103/RevModPhys.77.137

```python
# Instantiate a random graph and transform into an adjacency matrix
n_nodes = 500
graph_nx = nx.erdos_renyi_graph(n=n_nodes, p=1) # p=1 -> all-to-all connectivity
graph = nx.to_numpy_array(graph_nx)

# Run model with different coupling (K) parameters
coupling_vals = np.linspace(0, 0.6, 100)
runs = []
for coupling in coupling_vals:
    model = Kuramoto(coupling=coupling, dt=0.1, T=500, n_nodes=n_nodes)
    model.natfreqs = np.random.normal(1, 0.1, size=n_nodes)  # reset natural frequencies
    act_mat = model.run(adj_mat=graph)
    runs.append(act_mat)

# Check that natural frequencies are correct (we need them for prediction of Kc)
plt.figure()
plt.hist(model.natfreqs)
plt.xlabel('natural frequency')
plt.ylabel('count')
plt.tight_layout()
```




```python
# Plot all time series for all coupling values (color coded)
runs_array = np.array(runs)

plt.figure()
for i, coupling in enumerate(coupling_vals):
    plt.plot(
        [model.phase_coherence(vec)
         for vec in runs_array[i, ::].T],
        c=str(1-coupling),  # higher -> darker
    )
plt.ylabel(r'order parameter ($R_t$)')
plt.xlabel('time')
plt.tight_layout()
```




```python
# Plot final Rt for each coupling value
plt.figure()
for i, coupling in enumerate(coupling_vals):
    r_mean = np.mean([model.phase_coherence(vec)
                      for vec in runs_array[i, :, -1000:].T]) # mean over last 1000 steps
    plt.scatter(coupling, r_mean, c='steelblue', s=20, alpha=0.7)

# Predicted Kc - analytical result (from paper)
Kc = np.sqrt(8 / np.pi) * np.std(model.natfreqs) # analytical result (from paper)
plt.vlines(Kc, 0, 1, linestyles='--', color='orange', label='analytical prediction')

plt.legend()
plt.grid(linestyle='--', alpha=0.8)
plt.ylabel('order parameter (R)')
plt.xlabel('coupling (K)')
sns.despine()
plt.tight_layout()
```