

Chimera States in the Leaky Integrate-and-Fire Model with Non-Local Connectivity

Fotios Ioannis Giasemis



Project Submitted in Partial Fulfilment of the
Requirements for the Degree of
Master of Applied Mechanics

National Technical University of Athens
School of Applied Mathematical and Physical Sciences
June 30, 2021

Contents

1	Introduction	2
2	The leaky integrate-and-fire model	2
2.1	The single neuron model	2
2.2	Nonlocally coupled network of LIF neurons	3
3	Methods	4
4	Chimera states in coupled LIF neurons	4
4.1	Minimum number of neurons and $\Delta\omega$ vs N	4
4.2	Varying σ	5
4.3	Dependence on initial conditions	6
5	The modified LIF model	8
5.1	Chimeras states in the non-leaky integrate-and-fire model	8
5.2	Varying λ	9
6	Conclusions	11
7	Appendix: Python script	12
	References	14

Abstract

We study the dynamics of identical leaky integrate-and-fire neurons with symmetric non-local coupling. Upon varying control parameters such as the coupling strength and the coupling range, we investigate the system's behaviour and highlight the formation of chimera states. We also, examine a modified version of the model by varying the coefficient of the leak term.

1 Introduction

The study of dynamics and in particular collective behaviour of coupled oscillators has received great interest from scientists in different fields, from chemical systems to neuroscience and beyond. A particularly interesting and unexpected phenomenon observed in coupled oscillators is the so-called *chimera state*. In this dynamical scenario, some oscillators are synchronised while the others are not. These states were first observed in 2002 by Kuramoto and Battogtokh [1] while the term chimera was coined in 2004 by Abrams and Strogatz [2]. Potential applications of this phenomenon include the unihemispheric sleep observed in dolphins and some birds, power grids and social systems. This phenomenon has been observed numerically in various neuron models.

In this study we examine the effect of different control parameters on the appearance of chimera states for leaky integrate-and-fire (LIF) neuronal oscillators arranged in a regular ring topology.

2 The leaky integrate-and-fire model

2.1 The single neuron model

The LIF model is a simple model for spiking neurons introduced in 1907 by Louis Lapicque [3]. The main dynamical variable is the membrane potential $u(t)$, which evolves according to the equation

$$\frac{du(t)}{dt} = \mu - \lambda u(t), \quad (1)$$

with a reset condition

$$\forall u(t) = u_{th} \Rightarrow \lim_{\epsilon \rightarrow 0} u(t + \epsilon) = u_{rest}, \quad (2)$$

where u_{th} is the threshold of the potential, u_{rest} is the membrane potential at the rest state, and $\mu > u_{th}$ is a constant. The parameter λ is set to $\lambda = 1$ for the original LIF model, but it is included here because in the last section of this paper we study a modified version of the model.

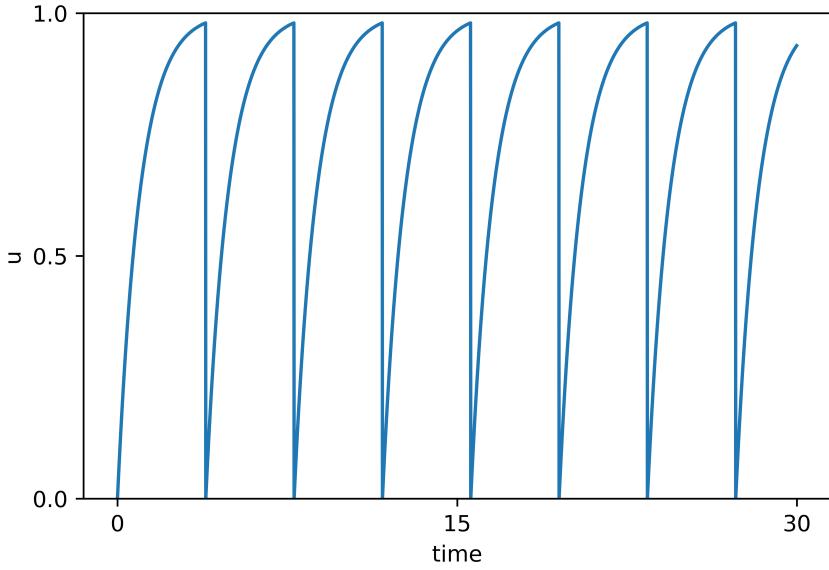


Figure 1: Dynamic evolution of the membrane potential in time according to Eq. (1) and (2).

The model, Eq. (1), includes a leak term, appearing as $-u(t)$ on the right-hand side, and is linear. However, with the addition of the reset condition, Eq. (2), the model becomes highly non-linear. Whenever the potential reaches the threshold u_{th} , it drops to u_{rest} . The resting potential can be set to $u_{rest} = 0$.

2.2 Nonlocally coupled network of LIF neurons

Neurons “fire” electrical signals as a result of receiving input from other neurons. Therefore, it is interesting to study networks of coupled neurons. We study a newtork of N LIF neurons that are arranged in a regular ring topology with non-local connections. That is, each element is coupled to its R nearest neighbours on either side. The dynamical evolution of this system is determined by

$$\frac{du_i(t)}{dt} = \mu - \lambda u_i + \frac{\sigma}{2R} \sum_{j=i-R}^{j=i+R} [u_i(t) - u_j(t)], \quad (3)$$

with the same reset mechanism for each element as in Eq. (2). Here, σ is the coupling strength and R is the coupling range. The index i has to be taken modulo N due to the topology and all the nodes are consider identical.

The study of a system of coupled oscillators involves the identification of parameter regions where synchronisation occurs. In the next sections, we investigate the effect of the coupling strength σ and the coupling range R on synchronisation phenomena with special focus on chimera states.

3 Methods

The calculations were performed on a conventional PC and coded in Python (Python version 3.8.5) and the total processing time was ≈ 240 hours. The code was verified by reproducing some of the results in [3], [4] and [5]. All the scripts used are based on variations of the code in the Appendix. The method used to integrate the ODE is a forward Euler method, with a step size of $dt = 0.01$.

4 Chimera states in coupled LIF neurons

We investigate the appearance of chimera states in a network of neurons without refractory period. The model used is the original LIF model, i.e. Eq. (1) and (2) with $\lambda = 1$.

4.1 Minimum number of neurons and $\Delta\omega$ vs N

We study systems with $N = 10, 20, 30, \dots, 100$, with $r = R/N = 0.4$ and $\sigma = 0.7$. Some snapshots are shown in Fig. (2) and (3).

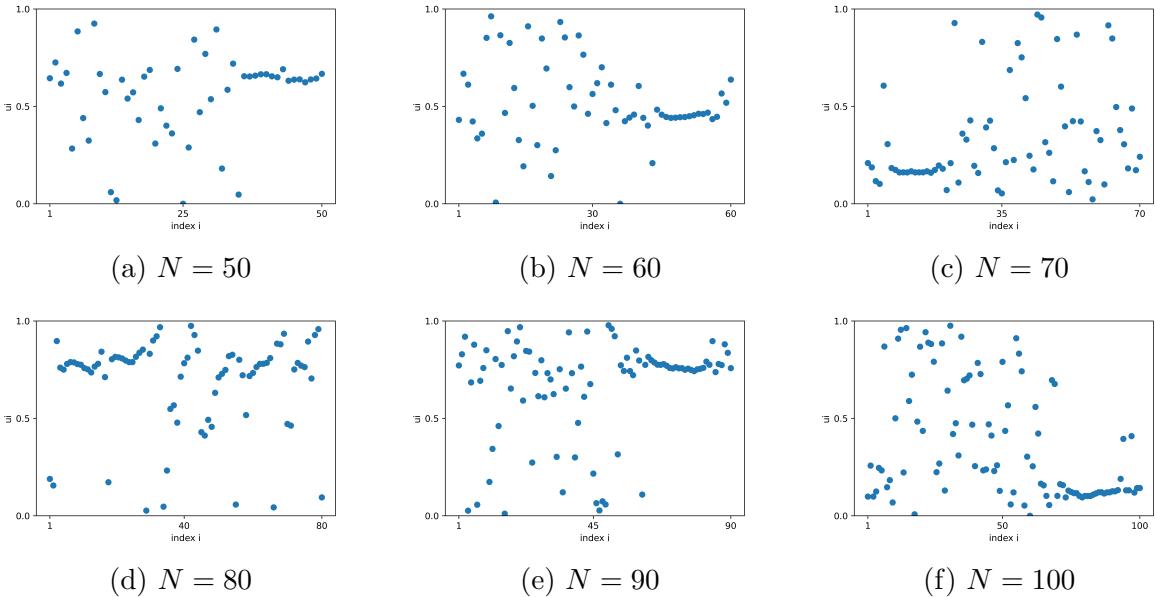


Figure 2: Snapshots of the membrane potential u_i at $t = 1000$ time units, for $\sigma = 0.7$, $r = 0.40$ and for various values of N .

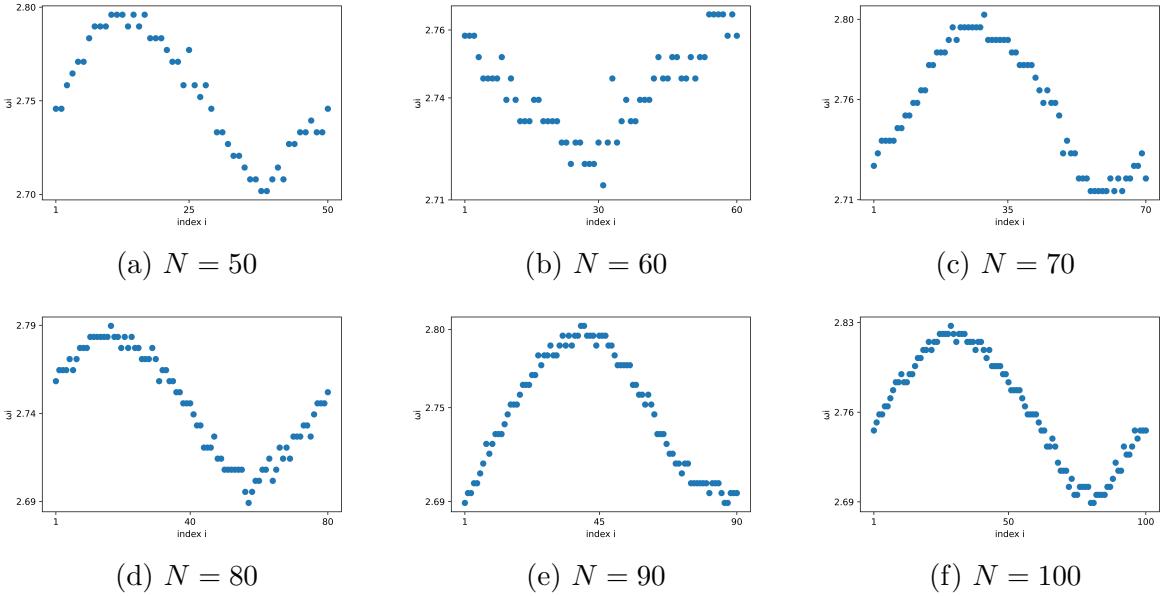


Figure 3: Mean phase-velocity profiles ω_i at $t = 1000$ time units, for $\sigma = 0.7$, $r = 0.40$ and for various values of N , as in Fig. (2).

We can clearly see chimera states for all N , but the smallest value of N for which the ω_i curve is smooth enough is for $N = 70$. Also, we have to compare $\Delta\omega = \omega_{\max} - \omega_{\min}$. In Fig. (4), $\Delta\omega$ vs N is shown.

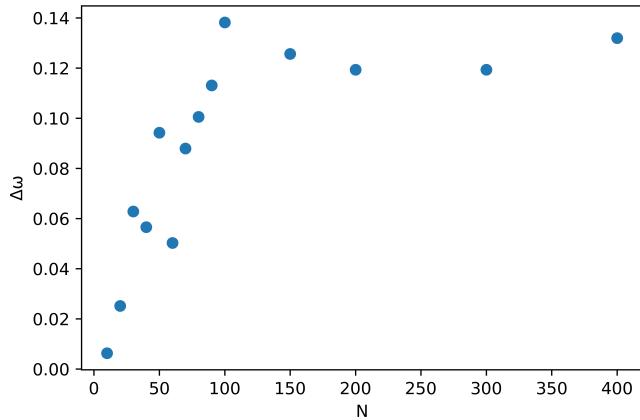


Figure 4: $\Delta\omega = \omega_{\max} - \omega_{\min}$ with varying system size N .

We conclude that the minimum number to use is $N = 70$. In other words, the computationally cheapest system is with $N = 70$.

4.2 Varying σ

We then vary σ for $0.1 \leq \sigma \leq 2.0$, for fixed $r = 0.4$ and $N = 70$. We observe chimera states around $\sigma = 0.7$ as before, but also at large values of σ . In particular, for $\sigma \gtrsim 1.4$ we observe double chimeras (Fig. (5)). In Fig. (6), $\Delta\omega$ vs σ is shown.

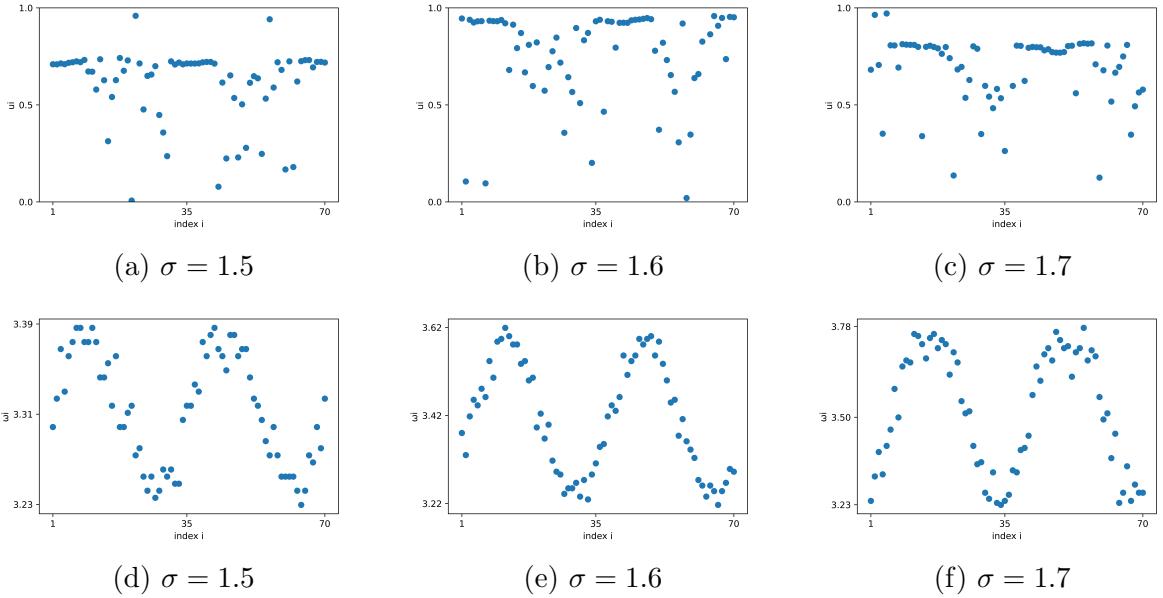


Figure 5: Snapshots of the membrane potential u_i and mean phase-velocity profiles ω_i for the double chimeras observed, for $N = 70$, $r = 0.40$ at $t = 1000$. The first row are the u_i and the second row are the ω_i .

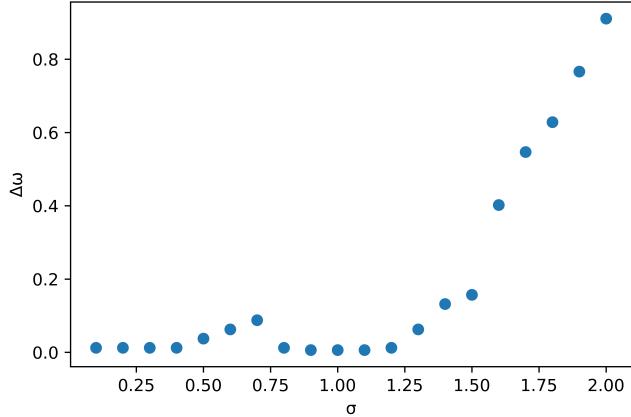


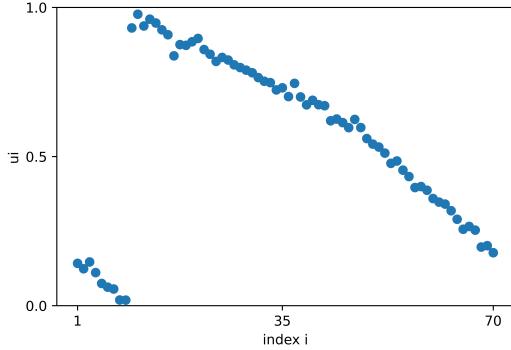
Figure 6: $\Delta\omega = \omega_{\max} - \omega_{\min}$ with varying σ .

The bump observed around $\sigma = 0.7$ is due to the appearance of chimeras around this value of σ . After $\sigma = 1.4$ we see $\Delta\omega$ to increase, which also agrees with the observation of double chimeras for these values.

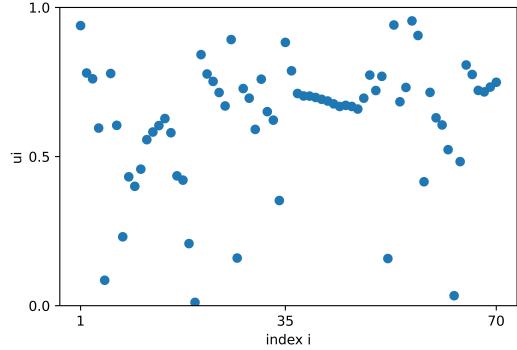
4.3 Dependence on initial conditions

We also investigate how the appearance of chimeras is dependent on the initial conditions. For $N = 70$, $\sigma = 0.7$ and $r = 0.35$, we run the system for different seed numbers with the command `numpy.random.seed()`. We save snapshots at $t = 1000$ and $t = 4000$

time units. First, we notice that chimeras are highly dependent on the initial conditions because there are seed numbers which result in chimeras (Fig. (7b)) while others that results in “chain-like” synchronisation (Fig. (7a)).



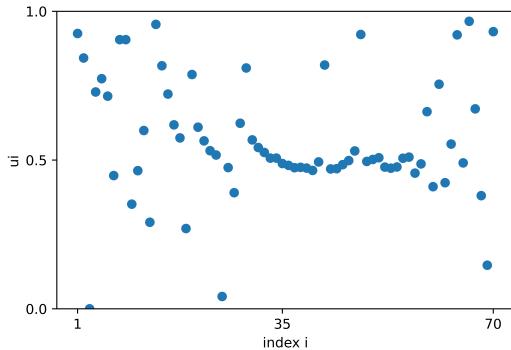
(a) seed = 64856



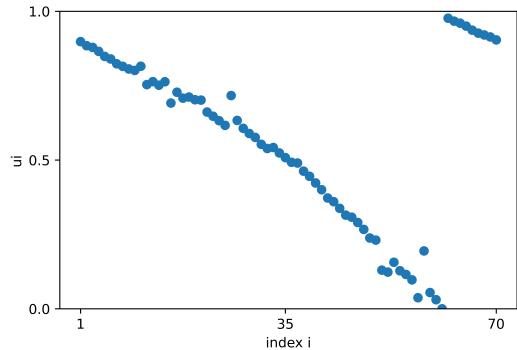
(b) seed = 38958

Figure 7: Snapshots of the membrane potential u_i at $t = 1000$, for $N = 70$, $\sigma = 0.7$ and $r = 0.35$, for two different initial conditions.

On the other hand, we have cases where chimeras are transient (Fig. (8)) and others that are not (Fig. (9)).



(a) $t = 1000$, seed = 70893



(b) $t = 4000$, seed = 70893

Figure 8: Snapshots of the membrane potential u_i , for $N = 70$, $\sigma = 0.7$ and $r = 0.35$, at $t = 1000$ and $t = 4000$.

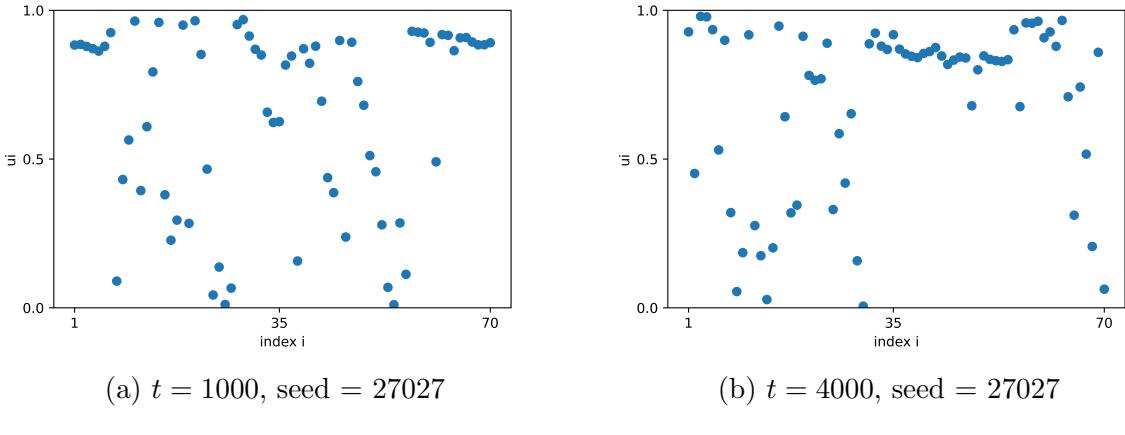


Figure 9: Snapshots of the membrane potential u_i at $t = 1000$ and $t = 4000$, for $N = 70$, $\sigma = 0.7$ and $r = 0.35$.

We note that the coherent (and incoherent) region of the chimera in Fig. (9) is moving slowly with time. These chimeras are called “travelling”.

5 The modified LIF model

We now study chimeras when we modify the LIF model by varying the parameter λ .

5.1 Chimeras states in the non-leaky integrate-and-fire model

We start with setting $\lambda = 0$. For $r = 0.35$ we vary σ from 0.1 to 2.0 in 0.1 increments, running each model for 1000 time units. We observe asynchronisation for $0.1 \leq \sigma \leq 0.8$ and $1.6 \leq \sigma \leq 2.0$. Therefore, we focus on the range $0.9 \leq \sigma \leq 1.5$. The most significant signature for a chimera is observed for $\sigma = 1.4$. For $N = 200$, the snapshots for $t = 1000$ and $t = 4000$ time units are shown in Fig. (10). The respective ω_i profile for $t = 4000$ is also shown in Fig. (11).

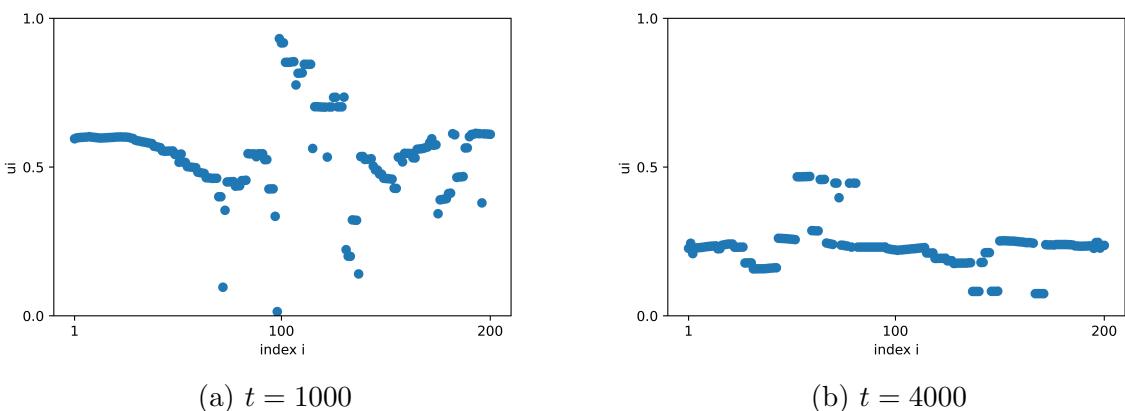


Figure 10: Snapshots of the membrane potential u_i at $t = 1000$ and $t = 4000$, for $N = 200$, $\sigma = 1.1$ and $r = 0.35$.

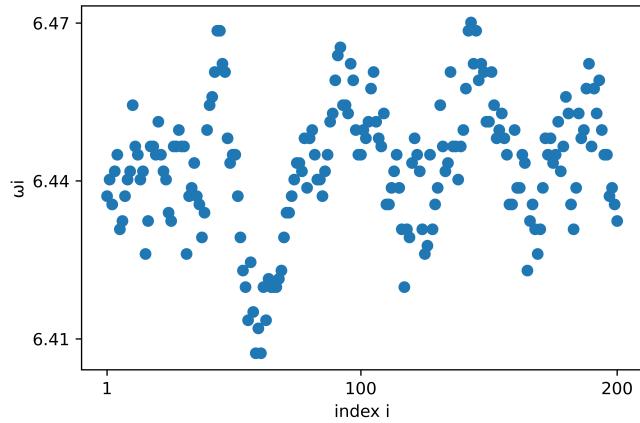


Figure 11: Mean phase-velocity profile ω_i at $t = 4000$, for $N = 200$, $\sigma = 1.1$ and $r = 0.35$.

5.2 Varying λ

We now choose a set of parameters where a “strong” chimera is observed in the original $\lambda = 1$ LIF model. We slowly vary λ , in order to study how the chimera disintegrates due to the changing λ . For $N = 150$, $r = 0.4$ and $\sigma = 0.7$, the chimera disintegrates almost completely at around $\lambda = 0.95$. In Fig. (12), we see how $\Delta\omega$ changes when λ is varied. Already at $\lambda = 0.95$, $\Delta\omega$ drops to $\Delta\omega \approx 0.04$. In Fig. (13), snapshots of the potential u_i are shown. The respective phase-velocity profiles ω_i are shown in Fig. (14).

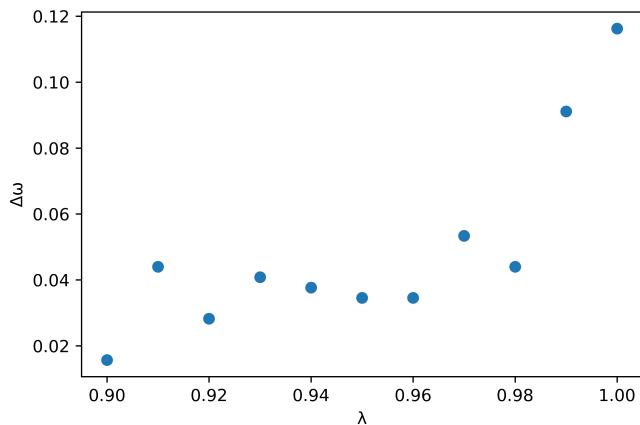


Figure 12: $\Delta\omega = \omega_{\max} - \omega_{\min}$ with varying λ .

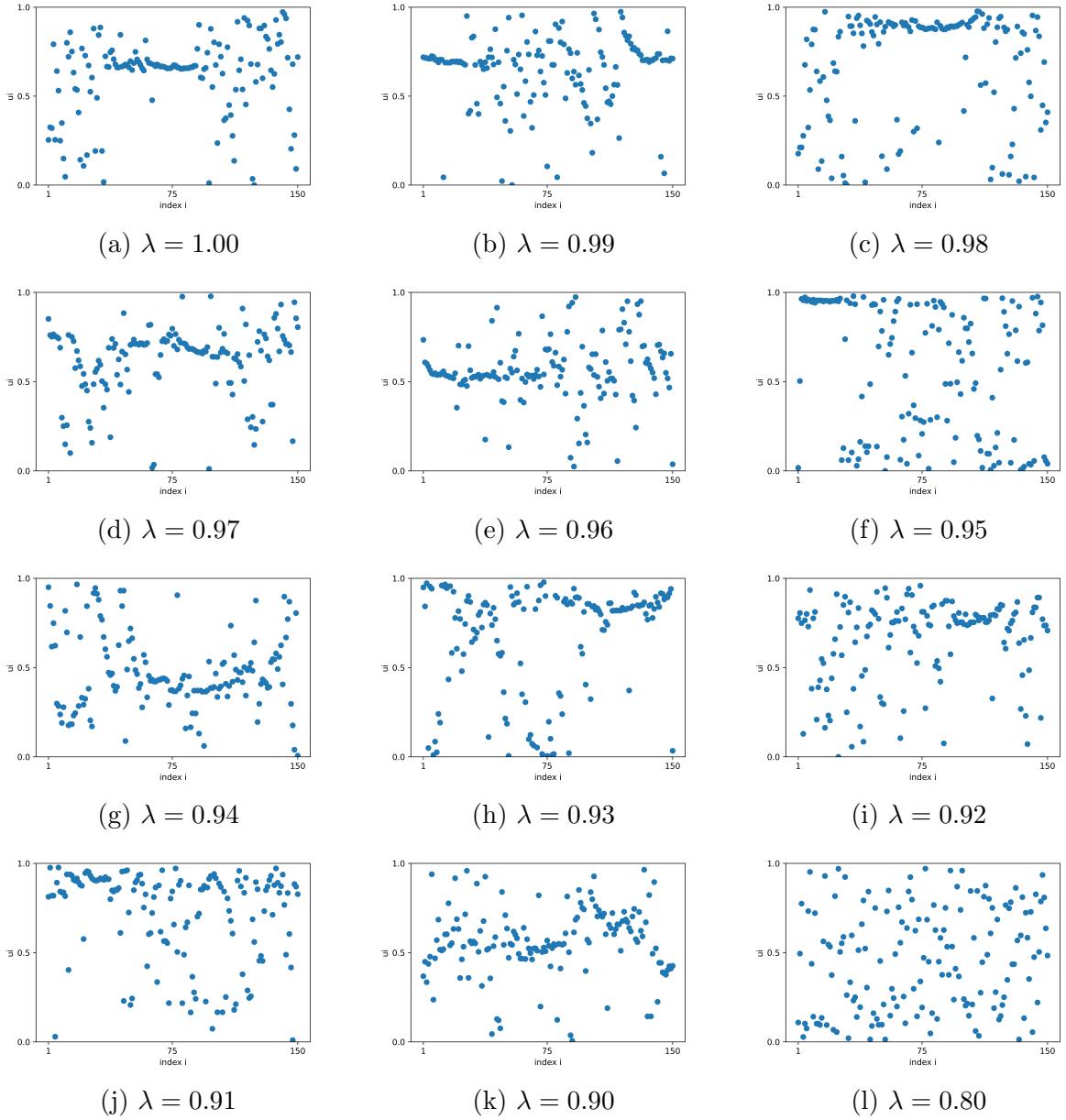


Figure 13: Snapshots of the membrane potential u_i at $t = 2000$ time units, for $N = 150$, $r = 0.4$ and $\sigma = 0.7$ and for various values of λ .

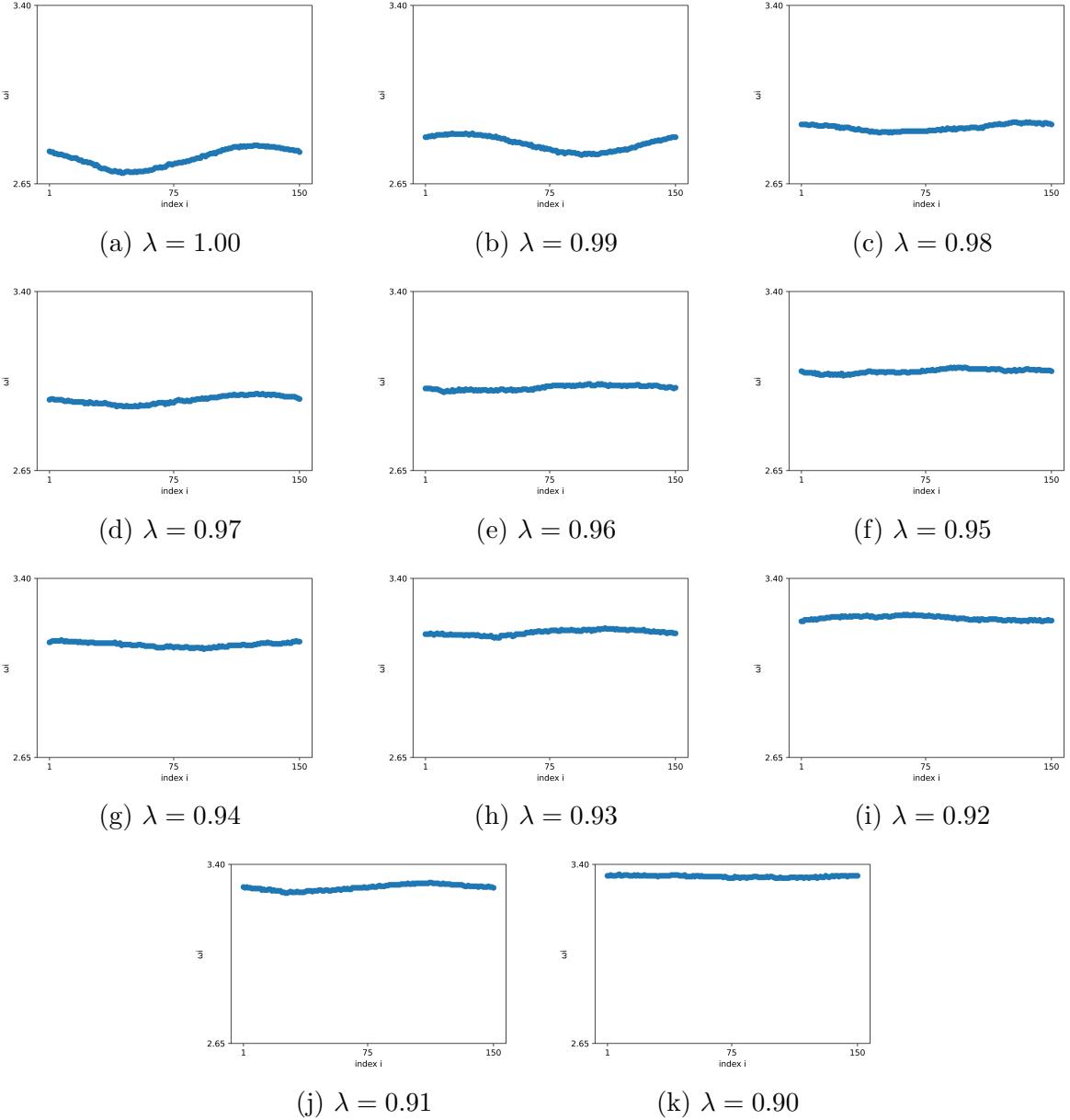


Figure 14: Mean phase-velocity profiles ω_i at $t = 2000$ time units, for $N = 150$, $r = 0.4$ and $\sigma = 0.7$ and for various values of λ .

6 Conclusions

Chimera states on a non-locally coupled network of LIF neurons highly depends on the coupling strength and the coupling range. The analysis of networks of around $N = 100$ neurons has shown the appearance of chimeras for intermediate values of coupling strength, for a large enough value of the coupling range. In addition, we observed that when we double the coupling strength, the single chimera becomes a double chimera. Finally, we observed that chimeras, as well as their duration, are highly dependent on initial conditions.

A similar investigation can be done to include the effect of a refractory period for the neuron. For example, it has been shown [3] that the refractory period helps chimeras survive for longer periods and is responsible for the formation of chimeras with multiple coherent and incoherent regions. Future work should address quantitative investigation of the parameter regions which favour chimera states and could include additional parameters related to the experimentally measured time-scales of biological neurons.

7 Appendix: Python script

```

1 # LIF model
2 # Nonlocal connectivity
3 # F. I. Giasemis
4
5 import numpy as np
6 import time
7 from matplotlib import pyplot as plt
8 from numpy import asarray
9 from numpy import savetxt
10 import csv
11 from notify_run import Notify
12
13 notify = Notify()
14
15 def run(N, R, sigma, time_max):
16     mu = 1
17     u_th = 0.98
18     dt = 0.01
19
20     seed_num = 38958
21     np.random.seed(seed_num)
22     u = np.random.rand(N)*u_th
23     circles = np.zeros(N)
24
25     t = 0
26     num = 0
27     plotfreq = 5000
28
29     start_time = time.time()
30
31     print('-----')
32     print('Starting calculation')
33     print('-----')
34
35     while t < time_max:
36         for i in range(N):
37             u[i] = u[i] + dt*(mu - u[i])
38             coupling = 0 # R <= N/2.
39             if i-R < 0:
40                 for j in range((i-R)%N,N):
41                     coupling = coupling + sigma*(u[i]-u[j])/(2*R)
42
43             for j in range(i+R+1):
44                 coupling = coupling + sigma*(u[i]-u[j])/(2*R)

```

```

45
46     u[i] = u[i] + dt*coupling
47     elif i+R > N-1:
48         for j in range(i-R,N):
49             coupling = coupling + sigma*(u[i]-u[j])/(2*R)
50
51     for j in range((i+R)%N+1):
52         coupling = coupling + sigma*(u[i]-u[j])/(2*R)
53
54     u[i] = u[i] + dt*coupling
55     else:
56         for j in range(i-R,i+R+1):
57             coupling = coupling + sigma*(u[i]-u[j])/(2*R)
58
59     u[i] = u[i] + dt*coupling
60
61     if u[i] > u_th:
62         u[i] = 0
63         circles[i] += 1
64
65     if num%plotfreq == 0:
66         print('t = ', round(t))
67         plt.scatter(range(1,N+1), u)
68         plt.ylim([0, 1])
69         plt.yticks([0, 0.5, 1])
70         plt.ylabel("ui")
71         plt.xticks([1, N/2, N])
72         plt.xlabel("index i")
73         plt.show()
74
75     t = t + dt
76     num = num + 1
77
78     print('-----')
79     print('End of calculation')
80     print('-----')
81     time_calc = time.time() - start_time
82     print("Time: %s seconds." % (round(time_calc, 2)))
83     print('-----')
84
85     """
86     Saving data.
87
88     """
89     w = 2*np.pi*circles/t
90     plt.scatter(range(1, N+1), w)
91     plt.ylabel("wi")
92     plt.yticks([round(min(w),2), round((min(w)+max(w))/2,2), round(max(w),2)])
93     plt.xticks([1, N/2, N])
94     plt.xlabel("index i")
95     plt.savefig('data/w_sigma=%s.png' % sigma, dpi=800)
96     plt.show()
97
98     plt.scatter(range(1,N+1), u)

```

```

99     plt.ylim([0, 1])
100    plt.yticks([0, 0.5, 1])
101    plt.ylabel("ui")
102    plt.xticks([1, N/2, N])
103    plt.xlabel("index i")
104    plt.savefig('data/u_sigma=%s.png' % sigma, dpi=800)
105    plt.show()
106
107    delta_w = max(w) - min(w)
108
109    with open('data/deltaw_data.csv', 'a', newline='') as file:
110        writer = csv.writer(file)
111        writer.writerow([delta_w])
112
113    u_data = asarray(u)
114    w_data = asarray(w)
115    savetxt('data/u_data_sigma=%s.csv' % sigma, u_data, delimiter=',')
116    savetxt('data/w_data_sigma=%s.csv' % sigma, w_data, delimiter=',')
117    with open('data/calc_sigma=%s.txt' % sigma, 'w') as f:
118        f.write('Results:')
119        f.write('\n')
120        f.write('delta_w = %s' % delta_w)
121        f.write('\n')
122        f.write('time_calc = %s' % time_calc)
123        f.write('\n')
124        f.write('\n')
125        f.write('Parameters:')
126        f.write('\n')
127        f.write('N = %s' % N)
128        f.write('\n')
129        f.write('R = %s' % R)
130        f.write('\n')
131        f.write('sigma = %s' % sigma)
132        f.write('\n')
133        f.write('time_max = %s' % time_max)
134        f.write('\n')
135        f.write('seed_num = %s' % seed_num)
136
137    notify.send('Calculation with sigma=%s ended.' % sigma)

```

References

- [1] Y. Kuramoto and D. Battogtokh. Coexistence of coherence and incoherence in non-locally coupled phase oscillators. *Nonlin. Phen. in Complex Sys.*, 5:380-385, 2002.
- [2] D. M. Abrams, S. H. Strogatz. Chimera states for coupled oscillators. *Phys. Rev. Lett.*, 93:174102, 2004.
- [3] N. D. Tsigkri-DeSmedt, J. Hizanidis, P. Hövel, A. Provata. Multi-chimera states and transitions in the leaky integrate-and-fire model with nonlocal and hierarchical connectivity. *Eur. Phys. J. Spec. Top.*, 225:1149–1164, 2016.
- [4] N. D. Tsigkri-DeSmedt, J. Hizanidis, P. Hövel, A. Provata. Multi-chimera states in the leaky integrate-and-fire model. *Procedia Computer Sci.*, 66:13-22, 2015.
- [5] N. D. Tsigkri-DeSmedt, J. Hizanidis, E. Schöll, A. Provata. Chimeras in leaky integrate-and-fire neural networks: effects of reflecting connectivities. *Eur. Phys. J. B*, 90:139, 2017.