ABSTRACT

COSS, OWEN T. Analyzing the Equilibria of Coupled Oscillators: Finding, Stability of, and Counting Equilibria for the Generalized Kuramoto Model. (Under the direction of Hoon Hong.)

Systems of coupled oscillators are prevalent throughout nature and engineering. These systems often exhibit complex synchronization phenomena and thus can be difficult to model. The most popular models for such systems are the Kuramoto model and its nonuniform coupling generalization that uses a symmetric matrix to describe the coupling. Studying the equilibria of these models yields insights into the synchronization behavior of systems of coupled oscillators.

This paper examines the equilibria in three ways. First, methods are presented that find all of the equilibria or classify the structure of all of the equilibria. Second, the stabilities of the equilibria are studied. Third, upper bounds for the maximum number of equilibria are proved. This paper is organized around these three topics for a few different cases of the Kuramoto model and its generalized version which are as follows.

In the case where the oscillators are identical and coupled uniformly, the equilibria can be classified into two sets, balanced and unbalanced. Moreover, one can find all of the eigenvalues related to each equilibrium which shows that only one equilibrium is stable. For a system with two or three oscillators, there is a finite number of equilibria, but a system with four or more oscillators has an infinite number of equilibria. When there are four oscillators, all of the equilibria reside in a finite number of connected components, and systems with five or more oscillators are conjectured to also have a finite number of connected components of equilibria.

The next case considered is the generalized Kuramoto model where the coupling is described by a symmetric rank one matrix. When the oscillators are identical, the equilibria can again be classified into two groups, balanced and unbalanced, and again only one equilibrium is stable. In the case where the oscillators are instead not uniform, an efficient algorithm is provided that computes all of the equilibria. Moreover, in experiments it compares favorably to other methods used to compute all the equilibria. This algorithm is used to show that generically there is a unique stable equilibrium for this case. It also provides a tighter upper bound on the number of equilibria than previously known.

Lastly, the techniques used for the rank one coupling case are applied to the generalized Kuramoto model where the coupling is described by a symmetric matrix of arbitrary rank. This provides an algorithm for finding all the equilibria with an efficiency that scales with the rank of the coupling matrix. This algorithm can also be used to more efficiently find only the stable equilibria provided the coupling matrix is positive semidefinite.

Two appendices are also provided. The first provides code examples implementing the equilibrium finding algorithms. The second shows that the methods used for the arbitrary rank case can also be applied on an even more general version of the Kuramoto model that includes a loss term to create an algorithm that finds all the equilibria. This is a potential topic for future research.

Analyzing the Equilibria of Coupled Oscillators: Finding, Stability of, and Counting Equilibria for the Generalized Kuramoto Model

by
Owen T. Coss

A dissertation submitted to the Graduate Faculty of
North Carolina State University
in partial fulfillment of the
requirements for the Degree of
Doctor of Philosophy

Mathematics

Raleigh, North Carolina

2020

APPROVED BY:

Seth Sullivant	Cynthia Vinzant
Jonathan Hauenstein	Hoon Hong Chair of Advisory Committee

BIOGRAPHY

Owen was born and raised in Greenville, South Carolina and is the oldest of three children. Growing up, he enjoyed playing games with his family and friends as well as reading fiction. In high school, he became fascinated by mathematics because of its elegant solutions to seemingly intractable problems. For his undergraduate education, he attended Bob Jones University, a Christian liberal arts university that is also in Greenville, and graduated with a major in Mathematics and Minor in Computer Science. He then started graduate studies in Mathematics at North Carolina State University.

ACKNOWLEDGEMENTS

Thank you to my parents and to my many teachers without whom this would not have been possible.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER

1

INTRODUCTION

Systems of coupled oscillators are common throughout nature and engineering. In fact, anything that continuously cycles through a fixed set of states or positions can be considered an oscillator. For example, fireflies that periodically flash, alternating current electricity, and metronomes are all oscillators. Often many such oscillators are coupled together, which means that the state of one oscillator can influence the speed of the oscillations of other oscillators in the group. In such situations, synchronization often occurs among the members. A common example is that a group of fireflies when left undisturbed in a forest will after a time start flashing nearly in unison [22]. In another example, if multiple metronomes are placed on a table which is allowed to sway, then the beat of each metronome will synchronize after a short time [44].

Because of the wide spread nature of such synchronization phenomena, it is important to have useful models so that this behavior can be studied. The most popular and successful model of this type is the Kuramoto model. This paper deals with three areas related to this model. First, methods of finding all the equilibria of the model or of classifying all the equilibria are examined in various situations. Second, stability properties of these equilibria are analyzed. And third, upper bounds on the number of equilibria are proved.

This paper is organized as follows. The remainder of Chapter one explains the Kuramoto model and a generalized variant, as well as explaining the terminology associated with the model. It also reviews previous results that will be needed in proofs in later chapters and reviews the results that compare to those presented in this paper.

Chapters two through five examine different cases of the model and its matrix coupling generalization. Specifically, Chapter two examines the original model in the special case where the oscillators are identical. In this case we can completely characterize all of the equilibria and their

associated eigenvalues, and we examine the number of connected components among the set of all equilibria. Chapters three and four examine the generalized Kuramoto model where the coupling matrix is rank one. Specifically, Chapter three deals with the case where the oscillators are all identical, whereas Chapter four deals with cases having nonidentical oscillators. When the oscillators are identical, there are two types of equilibria, and we show that there is a unique stable equilibria. In the nonidentical oscillator case, an algorithm is provided which can efficiently find all of the equilibria. This algorithm is compared to other solving methods and is shown to solve many examples in significantly less time. Additionally, it is shown that in cases having at least one equilibria, generically there is a unique stable equilibria. Furthermore, this algorithm can be utilized to provide a stricter upper bound on the number of solutions for the rank one case than the general bound from other works. Chapter five utilizes the methods of Chapter four on the generalized model with arbitrary rank coupling. This again yields an algorithm for finding all the equilibria, though without further results it is does not offer advantages to other methods of solving. However, when combined with a restriction that the coupling matrix is positive semidefinite, this algorithm does offer potential efficiency advantages when used to find just the stable solutions.

There are also two appendices. Appendix A provides two code examples. Fully functioning C++ code is provided that implements the finding algorithms from Chapter four, and prototype Matlab code is provided that implements the finding algorithm for one case of the arbitrary rank finding algorithm from Chapter five. Appendix B applies the methods used in Chapter five to a further generalization of the Kuramoto model that includes a loss term. While it is interesting to note that the same approach can be applied to that case, practically it offers no meaningful results without further research.

1.1 The Kuramoto Model

In 1975, Yoshiki Kuramoto presented a model for coupled oscillators and showed some preliminary results [31]. The model proved to be popular due to hitting the sweet spot between tractability and effectiveness at modeling the complicated phenomena present in systems of coupled oscillators. The model, which took on his name, is as follows.

$$\frac{\mathrm{d}\theta_u}{\mathrm{d}t} = \omega_u - \frac{K}{n}\sum_{v=1}^{n}\sin(\theta_u - \theta_v) \qquad \text{for } \ u = 1, 2, \ldots, n \tag{1.1.1}$$

where $n \geq 2$ is the number of oscillators, $K > 0$ gives the coupling strength, $\theta(t)$ is the phase angles of the oscillators at time t, and ω is the natural frequencies of the oscillators. By using the sine function on the angle differences, Eq. 1.1.1 models synchronization behavior since oscillators that are “too fast" will be slowed by the sum of sine terms and oscillators that are “too slow" will accelerate. Figure 1.1 gives a graphical representation of a single oscillator when we picture it as traveling along the unit circle.

The Kuramoto model has found widespread use in many fields. For example the model or a

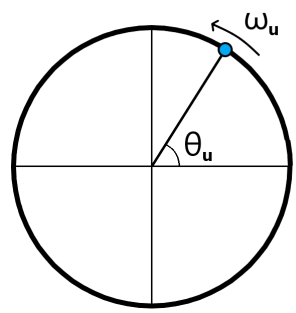


Figure 1.1 Representation of the uth oscillator on the unit circle. θ_u gives the phase angle and ω_u gives the natural frequency of the oscillator.

variant of it has been used for applications in electrical engineering [20, 21, 49], chemistry [3, 32, 43], physics [44], neurology [34, 47], and economics [27]. One of the most common generalizations allows for nonuniform coupling among the oscillators by using a coupling matrix $K \in \mathbb{R}_{\geq 0}^{n\times n}$.

$$\frac{\mathrm{d}\theta_u}{\mathrm{d}t} = \omega_u - \frac{1}{n}\sum_{v=1}^{n} K_{uv}\sin(\theta_u - \theta_v) \qquad \text{for } \ u = 1,2,\ldots,n \tag{1.1.2}$$

Typically the coupling matrix K is taken to be symmetric, but some more recent works have examined nonsymmetric coupling as well. Throughout the rest of this paper, the term “generalized Kuramoto model" is referring to this generalization, Eq. 1.1.2, with a symmetric coupling matrix.

1.1.1 Equilibria and the Rotating Reference Frame

Many important questions regarding the behavior of the Kuramoto model involve its equilibria. Since generally the oscillators in a coupled system will never stop moving, an equilibria in this context means an arrangement of the oscillators $\theta(t)$ such that

$$\frac{d\theta_1}{dt} = \cdots = \frac{d\theta_n}{dt} = E$$

for some constant E. In other words, a $\theta(t)$ for which all oscillators will have the same frequency. This is also sometimes referred to as a phase-locked state since the relative difference of the phases of the oscillators will be unchanging as they continue to rotate.

For such a $\theta(t)$, adding all n equations of Eq. 1.1.2 together gives

$$nE = \sum_{u=1}^{n}\omega_u - \frac{1}{n}\sum_{u=1}^{n}\sum_{v=1}^{n} K_{uv}\sin(\theta_u - \theta_v) = \sum_{u=1}^{n}\omega_u$$

Hence

$$E = \frac{1}{n}\sum_{u=1}^{n}\omega_u =: \bar{\omega}$$

where $\bar{\omega}$ is the average of $\omega_1, \omega_2, \ldots, \omega_n$. Thus the equation for an equilibrium becomes

$$0 = \bar{\omega}_u - \frac{1}{n}\sum_{v=1}^{n} K_{uv}\sin(\theta_u - \theta_v)$$

where $\bar{\omega}_u := \omega_u - \bar{\omega}$ measures the difference of each oscillator from the average. Note that $\sum_{u=1}^{n}\bar{\omega}_u = 0$. In order to simplify finding an equilibrium, it is thus equivalent to try to find $\theta(t)$ such that

$$\frac{\mathrm{d}\theta_1}{\mathrm{d}t} = \cdots = \frac{\mathrm{d}\theta_n}{\mathrm{d}t} = 0$$

with the restriction that

$$\sum_{u=1}^{n}\omega_u = 0$$

This technique is referred to as using a rotating reference frame and is how we will define equilibria. Furthermore, since $\frac{\mathrm{d}\theta_u}{\mathrm{d}t} = 0$, we have that $\theta(t)$ is a constant function in this reference frame. Thus we will drop t and use $\theta \in (-\pi, \pi]^n$. Note also that the same applies to the original Kuramoto model in Eq. 1.1.1 by letting every entry of the coupling matrix be the same constant.

Example 1.1.1 *Suppose three oscillators are coupled together with the natural frequencies of* 3.0, 3.1, *and* 3.5 *respectively. Then we have that* $\bar{\omega} = \frac{1}{3}(3.0+3.1+3.5) = 3.2$, *so we can use the rotating reference frame by letting*

$$\omega = (3.0-3.2,\ 3.1-3.2,\ 3.5-3.2) = (-0.2, -0.1, 0.3)$$

We will use the rotating reference frame to define equilibria and throughout the rest of the paper.

Definition 1.1.2 *θ is an **equilibrium** of Eq. 1.1.2 (or Eq. 1.1.1) under the rotating reference frame if*

$$\frac{d\theta_1}{dt} = \frac{d\theta_2}{dt} = \cdots = \frac{d\theta_n}{dt} = 0$$

1.1.2 Orbital Stability

In many applications, understanding the stability properties of an equilibrium is important. Since even in an equilibrium state the oscillators continue to move, we define stability relative to that movement. This idea is referred to as orbital stability. Note that if θ is an equilibrium of Eq. 1.1.2 (or Eq. 1.1.1), again under the assumption that we are using the rotating reference frame described last section, then any constant shift $\theta + c = (\theta_1 + c,\ \theta_2 + c, \ldots, \theta_n + c)$ for $c \in \mathbb{R}$ is also an equilibrium. However, since all the differences of phase angles between oscillators are identical for θ and $\theta + c$, we can consider these two solutions equivalent “modulo shift." Using this equivalence lets us define a notion of stability for the Kuramoto model.

Definition 1.1.3 *An equilibrium θ of Eq. 1.1.2 (or Eq. 1.1.1) is called **orbitally asymptotically stable** if there exists a δ-neighborhood centered at θ such that for any $\theta^{\delta}(t)$ starting within that neighborhood*

there exists a constant c such that

$$\lim_{t\to\infty} \|\theta^{\delta}(t)-(\theta+c)\| = 0$$

Put simply, an equilibrium θ is stable if after a small perturbation it returns to the same equilibrium modulo shift. Throughout the rest of the paper, “stable" will be taken to mean orbitally asymptotically stable.

One approach used in answering typical stability questions is to look at the eigenvalues of the Jacobian of a system when it is evaluated at an equilibrium. A similar approach works when considering orbitally stable solutions. Computing the Jacobian J for Eq. 1.1.2 gives

$$J = -\frac{1}{n}\begin{bmatrix} \sum_{v=1}^{n} K_{1v}\cos(\theta_1-\theta_v) & & \\ & \ddots & \\ & & \sum_{v=1}^{n} K_{nv}\cos(\theta_n-\theta_v) \end{bmatrix} + \frac{1}{n}\Big[K_{uv}\cos(\theta_u-\theta_v)\Big]_{(u,v)}$$

It is immediate that the sum of all the columns of J gives the zero vector. Thus J always has a zero eigenvalue with the associated eigenvalue being the all ones vector. This corresponds to the equivalence of equilibria under shift since adding the same constant amount to each oscillator doesn't change the angle differences between oscillators. Thus we can essentially ignore this eigenvalue and use the remaining $n-1$ eigenvalues to determine stability. Note also that J is symmetric, so all the eigenvalues of J are real. If J has $n-1$ negative eigenvalues at an equilibrium θ, then θ is orbitally asymptotically stable. Likewise, if J evaluated at an equilibrium has a positive eigenvalue, then that equilibrium is unstable [9].

1.2 Literature Review

This section reviews results dealing with the Kuramoto model. Because the model has been extensively studied since its inception in 1975 and many variations have been proposed for a wide variety of applications (e.g., adding in a loss term $\cos(\theta_u-\theta_v)$ or an inertia term $\frac{\mathrm{d}^2\theta_u}{\mathrm{d}t^2}$), only a selection of the vast number of papers will be covered. The results mentioned are those dealing with similar variations and questions to those in later chapters.

1.2.1 General Review

In this section a few general results are presented which will be required for proofs in later chapters. The first Lemma is a result from Chapter 7 of [5] which shows the affects on the eigenvalues of a matrix upon adding a positive semidefinite matrix to it. This result will be used in several of the stability results.

Lemma 1.2.1 *Let $A, B \in \mathbb{R}^{n\times n}$ be symmetric matrices with $B \succeq 0$. Then*

$$\lambda_u(A+B) \geq \lambda_u(A)$$

where $\lambda_1(M), \lambda_2(M), \ldots, \lambda_n(M)$ are the eigenvalues of the matrix M in decreasing order. Moreover, if $B \succ 0$, then the eigenvalues of $A+B$ are strictly greater than those of A.

The proof follows from the Courant-Fischer min-max theorem which is also presented in Chapter 7 of [5].

The next result is from [1] and gives a method to factor a determinant with a particular form. It will be used in the rank one stability results.

Lemma 1.2.2 *Let A be an $n \times n$ matrix and let b and c be n dimensional column vectors. Then*

$$\det\left(A + b\,b^{\mathrm{T}} + c\,c^{\mathrm{T}}\right) = \det(A)\left[\left(1 + b^{\mathrm{T}}A^{-1}b\right)\left(1 + c^{\mathrm{T}}A^{-1}c\right) - \left(c^{\mathrm{T}}A^{-1}b\right)^2\right]$$

Proof. Define the matrix

$$M := \begin{bmatrix} 1 & -b^{\mathrm{T}} & 0 \\ b & A & -c \\ 0 & c^{\mathrm{T}} & 1 \end{bmatrix}$$

Using elementary row and column operations on M, it is possible to convert it into two different lower triangular matrices M_1 and M_2.

$$M_1 := \begin{bmatrix} 1 & 0 & 0 \\ b & A + b\,b^{\mathrm{T}} + c\,c^{\mathrm{T}} & 0 \\ 0 & c^{\mathrm{T}} & 1 \end{bmatrix}$$

$$M_2 := \begin{bmatrix} 1 + b^{\mathrm{T}}A^{-1}b & 0 & 0 \\ b & A & 0 \\ 0 & c^{\mathrm{T}} & 1 - c^{\mathrm{T}}A^{-1}\left(\frac{b\,b^{\mathrm{T}}A^{-1}c}{1+b^{\mathrm{T}}A^{-1}b} - c\right) \end{bmatrix}$$

Using the fact that $\det(M_1) = \det(M_2)$ gives the result. ■

The main stability result in the arbitrary rank case relies on the Hadamard product of two matrices and the Schur Product Theorem which was first proven in [46].

Definition 1.2.3 *Let $A, B \in \mathbb{C}^{m\times n}$. Then the **Hadamard Product** of A and B, denoted $A \circ B$, is defined by $[A \circ B]_{uv} = A_{uv}B_{uv}$ for all $1 \leq u \leq m$, $1 \leq v \leq n$.*

Theorem 1.2.4 (Schur Product Theorem) *If A and B are square matrices of the same dimensions, then $A, B \succeq 0$ imples that $A \circ B \succeq 0$.*

Proof. Suppose $A, B \in \mathbb{C}^{n\times n}$ with $A, B \succeq 0$. Since A and B must be Hermitian matrices, we can

decompose them as

$$A = \sum_{u=1}^{n} \lambda_u a_u a_u^* \qquad\qquad B = \sum_{u=1}^{n} \mu_u b_u b_u^*$$

where $*$ is the conjugate transpose operator, λ is the set of eigenvalues of A and μ is the set of eigenvalues of B. Then we have that

$$\begin{aligned} A \circ B &= \sum_{u=1}^{n}\sum_{v=1}^{n} \lambda_u \mu_v (a_u a_u^*) \circ (b_v b_v^*) \\ &= \sum_{u=1}^{n}\sum_{v=1}^{n} \lambda_u \mu_v (a_u \circ b_v)(a_u \circ b_v)^* \end{aligned}$$

Since $\lambda_u \mu_v \geq 0$ for any u and v, each term of the sum is positive semidefinite, so $A \circ B$ is positive semidefinite. ■

1.2.2 Finding

One of the major questions arising from the Kuramoto model is how to find all the equilibria. Yoshiki Kuramoto analyzed this question in his original paper on the model [31] and in his follow up work [32]. He introduced the order parameters r and ψ which satisfied the relation

$$r e^{i\psi} = \frac{1}{n} \sum_{v=1}^{n} e^{i\theta_v}$$

where $i := \sqrt{-1}$. With these new variables, the model equations can be decoupled as

$$\frac{\mathrm{d}\theta_u}{\mathrm{d}t} = \omega_u + K r \sin(\psi - \theta_u)$$

By taking the limit as $n \to \infty$, and assuming a probability density on ω, he showed that equilibria exist as long as K is large enough, i.e., as long as the coupling is strong enough. The value for K at which equilibria are possible is called the critical coupling strength and is often denoted K_c. Many works since have focused on finding approximations and bounds on K_c. The paper by Dörfler and Bullo [19] summarizes many of these prior results and presents an explicit necessary and sufficient condition for the existence of equilibria.

In situations where the coupling is strong enough to admit solutions to the model, a few different approaches have been used to find all the equilibria. By using the trigonometric identity

$$\sin(\theta_u - \theta_v) = \sin(\theta_u)\cos(\theta_v) - \cos(\theta_u)\sin(\theta_v)$$

one can transform the original model into a polynomial system. Combined with the rotating refer-

ence frame explained in Section 1.1.1, the system for the equilibria is

$$\begin{cases} 0 = \omega_u - \dfrac{K}{n}\displaystyle\sum_{v=1}^{n} s_u c_v - c_u s_v \\ 1 = s_u^2 + c_u^2 \end{cases} \qquad \text{for } \; u = 1, 2, \ldots, n \tag{1.2.1}$$

This polynomial reformulation allows the use of standard algebraic geometry techniques such as resultants and Gröbner bases [7, 16, 17, 36]. However, these techniques scale very poorly as n grows. A typical example with $n = 11$ may take 40 hours to solve. Nonetheless, efficiency improvements have been made in some cases such as by using sparse elimination theory [48]. The Kuramoto model can also be rewritten in terms of Laurent polynomials by using $x_u = e^{i\theta_u}$ in which case the equilibria conditions become

$$0 = \omega_u - \frac{K}{2ni}\sum_{v=1}^{n}\left(\frac{x_u}{x_v} - \frac{x_v}{x_u}\right) \qquad \text{for } \; u = 1, 2, \ldots, n$$

This reformulation allows for certain sparse resultant methods to be used [8]. Focusing on a special case of the generalized model where the coupling matrix K is a tree network also allows for improved efficiency [10].

An alternate way to compute all of the equilibria using elliptical continuation was introduced in [35] which can be used on a more general class of models called the power-flow equations. It offers significant performance advantages compared to the previously mentioned techniques with a typical example having $n = 18$ taking 50 hours. (Due to the exponentially bad scaling of the above algebraic geometry techniques, a general case with $n > 12$ can not realistically be solved with them.) Furthermore, elliptical continuation scales with the number of real solutions to the polynomial system, i.e., equilibria, rather than n, so a case that has $n = 60$ but only two equilibria can be solved in 90 minutes. However, it was shown that this method did not always find all of the equilibria [40]. An improved version was created in [33] that handled the known counter-examples of the original, but it is not known whether this version is guaranteed to find all of the equilibria.

A third approach to find the equilibria is to use homotopy continuation methods. Optimizations for the more general class of models called the power flow equations was done in 1989 [45]. Later the numerical polynomial homotopy continuation (NPHC) method was also applied to the power flow equations [39]. A typical example with $n = 18$ may take 3 hours to solve. This is significantly faster than the other methods, however this approach does find all complex solutions to a polynomial reformulation of the Kuramoto model and then filters out the non-real solutions, so it also scales poorly with n. Currently a system with n in the lower twenties is the realistic limit for this approach as it now stands.

Many applications are interested in situations were the number of oscillators could be in the hundreds. Thus an important area of research is to continue to find more efficient algorithms for computing all the equilibria. Since many applications are expected to have relatively few equilibria, methods similar to elliptical continuation that scale with the number of equilibria instead of n are

likely going to be most effective.

1.2.3 Stability

Many applications are concerned with the stability properties of the equilibria of the Kuramoto model and whether or not there is a unique stable solution (in the orbitally stable sense). It was shown as early as 1972, prior to the formulation of the Kuramoto model in 1975, that the load-flow problem, a more general system of equations, could have multiple stable solutions [29]. Thus one approach is to find conditions under which existence or uniqueness properties hold, or conditions under which desirable properties of the stable equilibria are created. In one such approach, it was shown for the Kuramoto model that if the coupling was strictly greater than the critical coupling value, there is a unique stable solution [1]. Another work showed that the identical oscillator case, i.e., the case where $\omega = \vec{0}$ using the rotating reference frame, always has the globally asymptotically stable solution $\theta = \vec{0}$ [28]. Other works have studied the impact of the parameters K and ω on the solution space, such as [37] which examines the $n = 2, 3, 4$ cases and shows the affects of the size of K on stability of the equilibria. Similarly, [6] characterizes the set of ω values that admit stable equilibria and show that it is convex. A plot from this paper of the solutions by index when $n = 3$ is reproduced in Figure 1.2. Other works such as [14] look at asymptotically stable equilibria and consider the rate of approach.

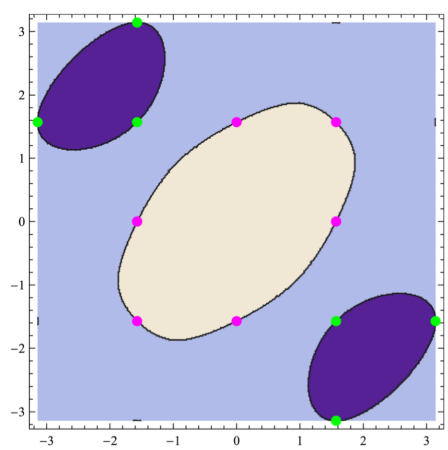


Figure 1.2 Plot of the index of equilibria when $n = 3$. The lighter central region contains the stable solutions, the pale blue region contains the solutions with index 1 Jacobian, and the purple regions contain the solutions with index 2 Jacobian. This plot is taken from [6].

Stability properties of generalizations of the Kuramoto model have also been studied. In [20], the authors consider the generalized Kuramoto model with non-identical coupling and provide a sufficient coupling strength condition for any arrangement of oscillators contained within a given arc to asymptotically approach a stable equilibrium. A parameter homotopy method was used in [38]

to experimentally find counterexamples to a few conjectures, and it also showed that multiple stable equilibria can exist in cyclic coupling situations. More exotic generalizations where the coupling is asymmetric or directed have also been studied. For instance [18] characterized the equilibria on acyclic oriented networks and oriented cyclic networks to show that if a stable equilibrium exists, it is unique. Overall, the number of stable equilibria depends heavily on the structure of the coupling.

1.2.4 Counting

A third topic of research is finding the maximum number of equilibria for the Kuramoto model. Applying the Bézout bound to the polynomial version of the Kuramoto model (see Eq. 1.2.1) gives an upper bound of 4^{n-1} on the number of complex solutions. Thus the equilibria, which correspond to the number of real solutions, are also bounded by 4^{n-1}. In 1982 Baillieul and Byrnes provided a tighter general upper bound [2]:

$$\binom{2n-2}{n-2} \approx \frac{4^{n-1}}{\sqrt{\pi(n-1)}}$$

When $n=2$ or $n=3$ it is possible to construct examples that show this bound is tight. For example [26] sampled the solution space in an $n=3$ case to graph the boundary curves between regions having different numbers of solutions. A color reproduction of this graph is provided in Figure 1.3. When $n=4$, tighter bounds have been found. It was shown in [41] that the power flow equations, a more general system of equations containing the Kuramoto model, has at most 16 real solutions provided one of the parameters is near zero, and examples are given which obtain this bound. In [50], again with $n=4$, it is shown that the number of distinct real solutions for the Kuramoto model is at most 14 and provides examples that have 10 distinct real solutions.

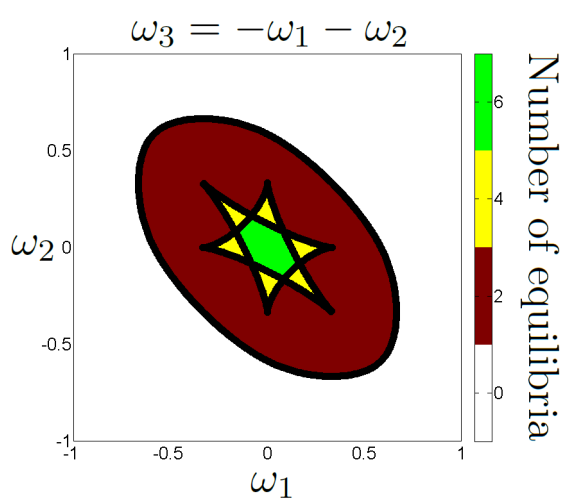


Figure 1.3 Regions based on the number of equilibria when $n=3$ and $K=1$. This graph is taken from [26].

Another approach used on the generalized Kuramoto model is to consider topology dependent bounds. In other words, one can interpret the coupling matrix K as a graph and determine bounds based on the structure of K. One of the more common ways used is to find the Bernstein-Khovanskii-Kushnirenko (BKK) bound on the polynomial version of the Kuramoto model (see Eq. 1.2.1) where K is replaced with a coupling matrix. A similar idea is used in [11] where the authors use the topology to construct an adjacency polytope which can be used to more tightly bound the number of solutions. Table 1.1 shows a comparison of the bounds from this method in two example network topologies in comparison to the Baillieul-Byrnes bound. One can also rewrite the Kuramoto model as a polynomial system similar to the Laurent polynomial version mentioned previously. Namely, by letting $x_u = e^{i\theta_u}$ and $y_u = e^{-i\theta_u}$, the equations for the equilibria using the rotating reference frame can be rewritten as

$$\begin{cases} 0 = \omega_u - \sum_{v=1}^{n} \frac{K_{uv}}{ni}\left(x_u y_v - x_v y_u\right) \\ 1 = x_u y_u \end{cases} \qquad \text{for } u = 1, 2, \ldots, n \tag{1.2.2}$$

This reformulation was used in [12] and it was shown that the BKK bound on this polynomial system improved on the BKK bound used on the typical polynomial version of the Kuramoto model in Eq. 1.2.1. A comparison of the two BKK bounds and the Baillieul-Byrnes bound is shown in Table 1.2. Another result in [13] uses the invariant intersection index to show that tree networks have an upper bound of 2^{n-1} and cycle networks have a bound of $n\binom{n-1}{\lfloor (n-1)/2 \rfloor}$. It is still an open question to find a tight bound for the generalized Kuramoto model for any topology of the coupling network.

Table 1.1 Comparison of bounds from [11] for various values of n. “Path AP" is the bound computed using the adjacency polytope on a path network graph, “Ring AP" is the bound computed using the adjacency polytope on a ring network graph, and “BB" is the Baillieul-Byrnes bound $\binom{2n-2}{n-1}$.

Bound $\backslash n$	4	5	6	7	8	9	10	11	12
Path AP	8	16	32	64	128	256	512	1024	2048
Ring AP	16	40	96	224	512	1152	2560	5632	12288
BB	20	70	252	924	3432	12870	48620	184756	705432

Table 1.2 Comparison of bounds on path and ring networks from [12] for various values of n. “BKK 1" is the BKK bound computed using Eq. 1.2.2, “BKK 2" is the BKK bound computed using the typical polynomial version of the Kuramoto model similar to Eq. 1.2.1, and “BB" is the Baillieul-Byrnes bound $\binom{2n-2}{n-1}$.

Bound $\backslash n$	4	5	6	7	8	9	10	11	12
Path BBK 1	8	16	32	64	128	256	512	1024	2048
Path BBK 2	24	80	256	832	2688	8704	28160	91136	294912
Ring BBK 1	12	30	60	140	280	630	1260	2772	5544
Ring BBK 2	24	80	256	832	2688	8704	28160	91136	294912
BB	20	70	252	924	3432	12870	48620	184756	705432

CHAPTER

2

UNIFORM COUPLING WITH IDENTICAL OSCILLATORS

The Kuramoto model with uniform all-to-all coupling is

$$\frac{\mathrm{d}\theta_u}{\mathrm{d}t} = \omega_u - \frac{K}{n}\sum_{v=1}^{n}\sin(\theta_u - \theta_v) \qquad \text{for } \; u = 1, 2, \ldots, n \tag{2.0.1}$$

where $n \geq 2$ is the number of oscillators, $K > 0$ is the coupling strength, θ is the phase angles of the oscillators, and ω is the natural frequencies of the oscillators. This chapter will focus on the case where the oscillators are identical, i.e., when $\omega_1 = \omega_2 = \cdots = \omega_n$. Using the rotating reference frame discussed in Section 1.1.1, this is equivalent to the case where $\omega_u = 0$ and $\frac{\mathrm{d}\theta_u}{\mathrm{d}t} = 0$ for every u. Thus the equations to solve to find the equilibria simplifies to

$$0 = \frac{K}{n}\sum_{v=1}^{n}\sin(\theta_u - \theta_v) \qquad \text{for } \; u = 1, 2, \ldots, n \tag{2.0.2}$$

2.1 Finding

Note that if θ is a solution to Eq. 2.0.1, then $\theta + c$ is also a solution for any constant c. Thus, two solutions are called equivalent modulo shift if each component-wise difference is the same modulo 2π. In order to choose a representative, an output condition for $\theta \in (-\pi, \pi]^n$ is given below so that only one of these equivalent solutions is chosen.

OC: $\quad \sum_{u=1}^{n} e^{i\theta_u} \in \mathbb{R}_+$ or $\left(\sum_{u=1}^{n} e^{i\theta_u} = 0 \text{ and } \theta_1 = 0\right)$

This choice of representative has a physical interpretation. If weights with masses given by K are placed on the unit circle in $\mathbb{C}$ at angles given by θ, then this choice of representative is the rotation where the center of mass lies on the positive real axis. This choice is always possible and unique provided that the center of mass is not exactly on the origin. In that situation, instead the first oscillator will be fixed to have a phase angle of zero.

Definition 2.1.1 *Let Θ_K be the set of solutions to Eq. 2.0.2 modulo shift. In other words,*

$$\Theta_K := \left\{ \theta \in (-\pi, \pi]^n : \underset{u}{\forall}\ 0 = \frac{K}{n} \sum_{v=1}^{n} \sin(\theta_u - \theta_v) \wedge \mathbf{OC} \right\}$$

Note that $\mathbb{U}$ is the unit circle in $\mathbb{C}$. In other words, $\mathbb{U} := \{z \in \mathbb{C}\colon\ |z| = 1\}$. We can perform a convenient transformation on Θ_K and break it into two disjoint sets that characterize the equilibria.

Theorem 2.1.2 *Let*

$$f : \left([\theta_1, \ldots, \theta_n]^{\mathrm{T}}\right) \longmapsto \left[e^{i\theta_1}, \ldots, e^{i\theta_n}\right]^{\mathrm{T}}.$$

Then we have

1. *$f|_{\Theta_K}$ is injective.*
2. *$f(\Theta_K) = \Phi_{K,unbalanced} \uplus \Phi_{K,balanced}$ where*

$$\Phi_{K,unbalanced} := \left\{ \phi \in \{-1, +1\}^n : \sum_{u=1}^{n} \phi_u > 0 \right\}$$
$$\Phi_{K,balanced} := \left\{ \phi \in \mathbb{U}^n : \sum_{u=1}^{n} \phi_u = 0 \wedge \phi_1 = 1 \right\}$$

Proof. Let $r := \frac{1}{n} \sum_{u=1}^{n} \phi_u$ where $\phi_u := e^{i\theta_u}$. Note that z^* denotes the complex conjugate of z. We have that

$$\begin{aligned}
f(\Theta_K) &= \left\{ f(\theta) : \underset{u}{\forall}\ 0 = \frac{K}{n} \sum_{v=1}^{n} \sin(\theta_u - \theta_v) \wedge \mathbf{OC} \right\} \\
&= \left\{ f(\theta) : \underset{u}{\forall}\ 0 = \frac{K}{n} \sum_{v=1}^{n} \mathrm{Im}\left(e^{i(\theta_u - \theta_v)}\right) \wedge \mathbf{OC} \right\} \\
&= \left\{ \phi \in \mathbb{U}^n : \underset{u}{\forall}\ 0 = \frac{K}{n} \sum_{v=1}^{n} \mathrm{Im}\left(\phi_u \phi_v^*\right) \wedge \left(r \in \mathbb{R}_+ \vee (r = 0 \wedge \phi_1 = 1)\right) \right\} \\
&= \left\{ \phi \in \mathbb{U}^n : \underset{u}{\forall}\ 0 = K\, \mathrm{Im}\left(\frac{1}{n} \sum_{v=1}^{n} \phi_u \phi_v^*\right) \wedge \left(r \in \mathbb{R}_+ \vee (r = 0 \wedge \phi_1 = 1)\right) \right\}
\end{aligned}$$

$$
\begin{aligned}
&= \left\{ \phi \in \mathbb{U}^n : \underset{u}{\forall}\ 0 = \operatorname{Im}\left(\phi_u \frac{1}{n} \sum_{v=1}^{n} \phi_v^* \right) \wedge \left(r \in \mathbb{R}_+ \vee (r = 0 \wedge \phi_1 = 1) \right) \right\} \\
&= \left\{ \phi \in \mathbb{U}^n : \underset{u}{\forall}\ 0 = \operatorname{Im}\left(\phi_u r^* \right) \wedge \left(r \in \mathbb{R}_+ \vee (r = 0 \wedge \phi_1 = 1) \right) \right\} \\
&= \left\{ \phi \in \mathbb{U}^n : \underset{u}{\forall}\ 0 = \operatorname{Im}(\phi_u) \wedge r \in \mathbb{R}_+ \right\} \uplus \left\{ \phi \in \mathbb{U}^n : r = 0 \wedge \phi_1 = 1 \right\} \\
&= \Phi_{K,\text{unbalanced}} \uplus \Phi_{K,\text{balanced}}
\end{aligned}
$$

■

Corollary 2.1.3 *For $\phi \in \Phi_{K,unbalanced} \uplus \Phi_{K,balanced}$, we have that*

$$
f^{-1}(\phi) = \left[\arg(\phi_1), \arg(\phi_2), \ldots, \arg(\phi_n) \right]^{\mathrm{T}} \in \Theta_K
$$

2.2 Stability

In this section we will examine all of the eigenvalues of the equilibria and show that there is one stable equilbirium (in the orbitally stable sense; see Section 1.1.2 for a discussion of orbital stability). Let J be the Jacobian for Eq. 2.0.1. A straightforward calculation gives

$$
J = -\frac{K}{n} \begin{bmatrix} \sum_{v=1}^{n} \cos(\theta_1 - \theta_v) & & \\ & \ddots & \\ & & \sum_{v=1}^{n} \cos(\theta_n - \theta_v) \end{bmatrix} + \frac{K}{n} \Big[\cos(\theta_u - \theta_v) \Big]_{(u,v)} \tag{2.2.1}
$$

Lemma 2.2.1 *The Jacobian matrix has a zero eigenvalue.*

Proof. It is immediate from Eq. 2.2.1 that the sum of all the columns of J is zero. Therefore the all ones vector is an eigenvector with the corresponding eigenvalue of zero. ■

The all ones eigenvector corresponds to the invariance under shift of the Kuramoto model. Adding the same amount to every phase angle does not change the differences, so we will ignore it when defining stability for the Kuramoto model. Also note that since J is a real symmetric matrix, all the eigenvalues are real. Thus if J evaluated at an equilibrium has $n-1$ negative eigenvalues, then that equilibrium is stable. Likewise, if J has a positive eigenvalue, then that equilibrium cannot be stable [9].

Remark 2.2.2 *For convenience, we will refer to $\phi \in \Phi_{K,unbalanced} \uplus \Phi_{K,balanced}$ as stable if the corresponding $\theta = f^{-1}(\phi)$ is stable.*

Note that if $\theta \in \Theta_K$, then $\theta' \in \Theta_K$ where θ' is a permutation of θ. Thus we would like to consider only one permutation of a solution when determining the stability. The following Lemma shows that this is valid since all permutations of a solution share the same eigenvalues.

Lemma 2.2.3 *Let θ' be a permutation of θ. Then the eigenvalues of the Jacobian for θ are the same as those for θ'.*

Proof. Let $p \in S_n$ be a permutation and let $\theta' = (\theta_{p_1}, \theta_{p_2}, \ldots, \theta_{p_n})$. Let J be the Jacobian matrix given in Eq. 2.2.1 using θ and J' be the Jacobian matrix given in Eq. 2.2.1 using θ'. Let P be the $n \times n$ permutation matrix that corresponds with p. Then we have that

$$J' = PJP^{-1}$$

Therefore J and J' have the same eigenvalues. ■

It is convenient for the proof later to convert the Jacobian into the same variables as the previous section used.

Lemma 2.2.4 *Let*

$$\phi_u := e^{i\theta_u} \qquad\qquad r := \frac{1}{n}\sum_{u=1}^{n}\phi_u$$

Then

$$J = -K\begin{bmatrix} \mathrm{Re}\left(\phi_1 r^\star\right) & & \\ & \ddots & \\ & & \mathrm{Re}\left(\phi_n r^\star\right) \end{bmatrix} + \frac{K}{n}\begin{bmatrix} \mathrm{Re}(\phi_1) & \mathrm{Im}(\phi_1) \\ \vdots & \vdots \\ \mathrm{Re}(\phi_n) & \mathrm{Im}(\phi_n) \end{bmatrix}\begin{bmatrix} \mathrm{Re}(\phi_1) & \cdots & \mathrm{Re}(\phi_n) \\ \mathrm{Im}(\phi_1) & \cdots & \mathrm{Im}(\phi_n) \end{bmatrix}$$

Proof. Applying the change of variables to Eq. 2.2.1 gives

$$J = -\frac{K}{n}\begin{bmatrix} \mathrm{Re}\left(\sum_{v=1}^{n}\phi_1\phi_v^\star\right) & & \\ & \ddots & \\ & & \mathrm{Re}\left(\sum_{v=1}^{n}\phi_n\phi_v^\star\right) \end{bmatrix} + \frac{K}{n}\Big[\mathrm{Re}(\phi_u\phi_v^\star)\Big]_{(u,v)}$$

$$= -K\begin{bmatrix} \mathrm{Re}\left(\phi_1 r^\star\right) & & \\ & \ddots & \\ & & \mathrm{Re}\left(\phi_n r^\star\right) \end{bmatrix} + \frac{K}{n}\Big[\mathrm{Re}(\phi_u\phi_v^\star)\Big]_{(u,v)}$$

Furthermore, note that the second matrix in the sum above can be factored.

$$J = -K\begin{bmatrix} \mathrm{Re}\left(\phi_1 r^\star\right) & & \\ & \ddots & \\ & & \mathrm{Re}\left(\phi_n r^\star\right) \end{bmatrix} + \frac{K}{n}\begin{bmatrix} \mathrm{Re}(\phi_1) & \mathrm{Im}(\phi_1) \\ \vdots & \vdots \\ \mathrm{Re}(\phi_n) & \mathrm{Im}(\phi_n) \end{bmatrix}\begin{bmatrix} \mathrm{Re}(\phi_1) & \cdots & \mathrm{Re}(\phi_n) \\ \mathrm{Im}(\phi_1) & \cdots & \mathrm{Im}(\phi_n) \end{bmatrix}$$

■

Remark 2.2.5 *This factorization of J can easily be done prior to the change of variable, i.e., on Eq. 2.2.1, with trigonometric identities, and that factorization is used in [1] and [6].*

Using Lemmas 2.2.3 and 2.2.4, we can find the eigenvalues for all the equilibria.

Theorem 2.2.6 *Let $\Phi_{K,unbalanced}$ and $\Phi_{K,balanced}$ be as in Theorem 2.1.2.*

- $\Phi_{K,unbalanced} := \left\{ \phi \in \{-1,+1\}^n : \sum_{u=1}^{n} \phi_u > 0 \right\}$

  Let p be the number of $+1$s in ϕ and m be the number of -1s in ϕ. Hence $p > m$. The eigenvalues for any $\phi \in \Phi_{K,unbalanced}$ are as follows.

  - $\lambda = (\underbrace{-K,\ldots,-K}_{n-1},\ 0)$ *if* $m = 0$.
  - $\lambda = (\underbrace{-d,\ldots,-d}_{p-1},\ 0,\ \underbrace{d,\ldots,d}_{m-1},\ K)$ *where* $d = \frac{p-m}{n}K$ *if* $m > 0$.

- $\Phi_{K,balanced} := \left\{ \phi \in \mathbb{U}^n : \sum_{u=1}^{n} \phi_u = 0 \wedge \phi_1 = 1 \right\}$

  The eigenvalues for any $\phi \in \Phi_{K,balanced}$ are as follows:

  - $\lambda = (\underbrace{0,\ldots,0}_{n-1},\ K)$ *if n is even and ϕ is any permutation of* $(\underbrace{1,\ldots,1}_{\frac{n}{2}},\ \underbrace{-1,\ldots,-1}_{\frac{n}{2}})$ *with* $\phi_1 = 1$.
  - $\lambda = (\underbrace{0,\ldots,0}_{n-2},\ a, K-a)$ *otherwise where*

$$a := \frac{K}{2}\left(1 - \sqrt{1 - \frac{4}{n^2} \sum_{1 \le u < v \le n} \mathrm{Im}\left(\phi_u \phi_v^*\right)^2}\right)$$

Remark 2.2.7 *The fact that $\theta = (0,\ldots,0)$, which corresponds to $\phi = (1,\ldots,1)$, is the only stable solution was previously shown in [1]. Moreover, stability properties of this equilibrium have been examined in [28].*

Proof. We will consider $\Phi_{K,\text{unbalanced}}$ and $\Phi_{K,\text{balanced}}$ separately.

- $\Phi_{K,\text{unbalanced}}$

  First, consider the case $\phi = (1,\ldots,1)$. Using $r := \frac{1}{n}\sum_{u=1}^{n} \phi_u$ gives $r = 1$ and the Jacobian from Lemma 2.2.4 becomes

$$J = \frac{K}{n}\begin{bmatrix} -n+1 & 1 & 1 & \cdots & 1 \\ 1 & -n+1 & 1 & \cdots & 1 \\ \vdots & & \ddots & & \vdots \\ 1 & \cdots & 1 & -n+1 & 1 \\ 1 & \cdots & 1 & 1 & -n+1 \end{bmatrix}$$

This matrix has the n eigenvectors

$$\begin{bmatrix} -1 \\ 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \begin{bmatrix} -1 \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \ldots, \begin{bmatrix} -1 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}, \text{and} \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix},$$

with the corresponding eigenvalues $-K, -K, \ldots, -K, 0$.

The other equilibria when converted to ϕ are some mixes of $+1$ and -1. Let p be the number of $+1$s and m be the number of -1s. By choosing the representative so that $r \in \mathbb{R}_+$, only the cases with $p > m$ need to be considered. Up to index permutations, all the equilibria have the form $\phi = (\underbrace{1, \ldots, 1}_{p}, \underbrace{-1, \ldots, -1}_{m})$. Furthermore, $r = \frac{\Delta}{n}$ where $\Delta := p - m$. Substituting these values into the Jacobian from Lemma 2.2.4 gives the following. (Note that the upper left submatrix is $p \times p$ and the lower right submatrix is $m \times m$.)

$$J = \frac{K}{n} \left[\begin{array}{ccc|ccc} -\Delta+1 & & 1 & & & \\ & \ddots & & & -1 & \\ 1 & & -\Delta+1 & & & \\ \hline & & & \Delta+1 & & 1 \\ & -1 & & & \ddots & \\ & & & 1 & & \Delta+1 \end{array}\right]$$

This results in four groups of eigenvectors as follows. (Note that the top section of each of the following eigenvectors have p elements, the bottom sections have m elements, and that $\vec{0}$ represents a column of zeros of the appropriate size.)

- $\left[\begin{array}{c} -1 \\ 1 \\ 0 \\ 0 \\ \vdots \\ 0 \\ \hline \vec{0} \end{array}\right], \left[\begin{array}{c} -1 \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \\ \hline \vec{0} \end{array}\right], \ldots, \left[\begin{array}{c} -1 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 1 \\ \hline \vec{0} \end{array}\right]$ give $p-1$ copies of the eigenvalue $-\frac{\Delta}{n}K$.
- $\begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}$ gives the eigenvalue 0.

- $\begin{bmatrix} \vec{0} \\ \hline 1 \\ -1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \begin{bmatrix} \vec{0} \\ \hline 1 \\ 0 \\ -1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \ldots, \begin{bmatrix} \vec{0} \\ \hline 1 \\ 0 \\ 0 \\ \vdots \\ 0 \\ -1 \end{bmatrix}$ give $m-1$ copies of the eigenvalue $\frac{\Delta}{n}K$.
- $\begin{bmatrix} -m \\ \vdots \\ -m \\ \hline p \\ \vdots \\ p \end{bmatrix}$ gives the eigenvalue K.

- $\Phi_{K,\text{balanced}}$

  Consider the Jacobian from Lemma 2.2.4 for the $r=0$ case.

$$J = \frac{K}{n} \begin{bmatrix} \mathrm{Re}(\phi_1) & \mathrm{Im}(\phi_1) \\ \vdots & \vdots \\ \mathrm{Re}(\phi_n) & \mathrm{Im}(\phi_n) \end{bmatrix} \begin{bmatrix} \mathrm{Re}(\phi_1) & \ldots & \mathrm{Re}(\phi_n) \\ \mathrm{Im}(\phi_1) & \ldots & \mathrm{Im}(\phi_n) \end{bmatrix}$$

  Therefore, J is at most rank 2, so there are at most 2 nonzero eigenvalues. Consider two subcases on the rank of J.

  - $\mathrm{Rk}(J)=1$

    Since $\phi_1 = 1$, we must have that $\mathrm{Im}(\phi_u)=0$ for all u in order for J to be rank one. The only possible way up to index permutations for this to occur is for n to be even and ϕ to be evenly split between $+1$ and -1, i.e., $\phi = (\underbrace{1,\ldots,1}_{\frac{n}{2}}, \underbrace{-1,\ldots,-1}_{\frac{n}{2}})$ so that $r=0$. This gives

$$J = \frac{K}{n} \left[\begin{array}{c|c} \mathbf{1} & -\mathbf{1} \\ \hline -\mathbf{1} & \mathbf{1} \end{array} \right]$$

where $\mathbf{1}$ is the $\frac{n}{2} \times \frac{n}{2}$ matrix filled with ones. We have that

$$\frac{K}{n}\left[\begin{array}{c|c} \mathbf{1} & -\mathbf{1} \\ \hline -\mathbf{1} & \mathbf{1} \end{array}\right]\begin{bmatrix} -1 \\ \vdots \\ -1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} = K \cdot \begin{bmatrix} -1 \\ \vdots \\ -1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}$$

Therefore the eigenvalues are $n-1$ copies of 0 and K.

- $\mathrm{Rk}(J) = 2$

  The characteristic polynomial of J is

$$\begin{aligned} C(\lambda) =& \lambda^n + c_{n-1}\lambda^{n-1} + c_{n-2}\lambda^{n-2} \\ =& \lambda^{n-2}(\lambda^2 + c_{n-1}\lambda + c_{n-2}) \end{aligned}$$

  Furthermore, $c_{n-1} = -\mathrm{tr}(J) = -K$ and c_{n-2} is the sum of the 2×2 principal minors of J, specifically

$$\begin{aligned} c_{n-2} &= \frac{K^2}{n^2} \sum_{1 \le u < v \le n} 1 - \mathrm{Re}(\phi_u \phi_v^\star)^2 \\ &= \frac{K^2}{n^2} \sum_{1 \le u < v \le n} \mathrm{Im}(\phi_u \phi_v^\star)^2 \end{aligned}$$

  since $|\phi_u \phi_v^\star| = 1$. Moreover, since $\mathrm{Rk}(J) = 2$, there exists a w such that $\mathrm{Im}(\phi_w) \neq 0$ and $w \neq 1$ since $\phi_1 = 1$. Thus $\mathrm{Im}(\phi_1 \phi_w)^2 > 0$, so $c_{n-2} > 0$. Consider the quadratic factor of the characteristic polynomial, $\lambda^2 - K\lambda + c_{n-2}$. We have

$$\lambda = \frac{K \pm \sqrt{K^2 - 4c_{n-2}}}{2} \tag{2.2.2}$$

  Since $c_{n-2} > 0$ and all the roots are real because J is a symmetric real matrix, the two roots of Eq. 2.2.2 are both positive and sum to K. Therefore the eigenvalues of J are $n-2$ copies of 0 and the two positive values satisfying Eq. 2.2.2 which have the form a and $K-a$ where $0 < a \le \frac{K}{2}$.

■

Example 2.2.8 *Let $n = 6$ and $K = 1$. Then Theorem 2.2.6 gives*

- $\Phi_{K,unbalanced}$ *has three elements up to index permutations, namely*
  - $\phi = (1,1,1,1,1,1)$ *with eigenvalues* $(-1,-1,-1,-1,-1,0)$

- $\phi = (1,1,1,1,1,-1)$ *with eigenvalues* $\left(-\frac{2}{3},-\frac{2}{3},-\frac{2}{3},-\frac{2}{3},0,1\right)$
- $\phi = (1,1,1,1,-1,-1)$ *with eigenvalues* $\left(-\frac{1}{3},-\frac{1}{3},-\frac{1}{3},0,\frac{1}{3},1\right)$

- $\Phi_{K,balanced}$ *has an infinite number of elements. Four examples are*
  - $\phi^{(1)} = (1,1,1,-1,-1,-1)$ *with eigenvalues* $(0,0,0,0,0,1)$
  - $\phi^{(2)} = (1,1,e^{i\frac{2\pi}{3}},e^{i\frac{2\pi}{3}},e^{i\frac{4\pi}{3}},e^{i\frac{4\pi}{3}})$ *with eigenvalues* $\left(0,0,0,0,\frac{1}{2},\frac{1}{2}\right)$
  - $\phi^{(3)} = (1,e^{i},i,-1,e^{i(\pi+1)},-i)$ *with eigenvalues* $\left(0,0,0,0,\frac{1}{3},\frac{2}{3}\right)$
  - $\phi^{(4)} = (1,e^{i(\alpha-\frac{\pi}{3})},e^{i(\alpha-\frac{\pi}{3})},e^{i(\frac{5\pi}{3}-\alpha)},e^{i(\frac{5\pi}{3}-\alpha)},e^{i\frac{2\pi}{3}})$ *where* $\alpha = \arccos\left(-\frac{1}{4}\right)$ *with eigenvalues* $\left(0,0,0,0,\frac{1}{8},\frac{7}{8}\right)$

Figure 2.1 has graphs of these four sample equilibria.

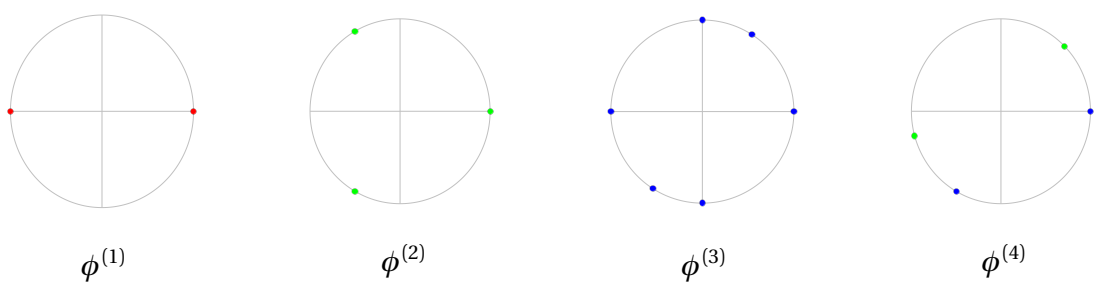


Figure 2.1 The four sample equilibria from $\Phi_{K,\text{balanced}}$ in Example 2.2.8. A blue circle represents one oscillator, green represents two, and red represents three.

2.3 Counting

Using Theorem 2.2.6 also allows us to count the number of equilibria for each index value.

Theorem 2.3.1 *Let N_u be the number of equilibria of index u modulo shift. (The index of an equilibrium is the number of positive eigenvalues of its associated Jacobian matrix.) Then*

$$N_0 = 1 \qquad\qquad N_2 = \begin{cases} 0 & \text{if } n=2 \\ 2 & \text{if } n=3 \\ \infty & \text{if } n\geq 4 \end{cases}$$

$$N_1 = \begin{cases} 1 & \text{if } n=2 \\ n & \text{if } n \text{ is odd} \\ n+\binom{n-1}{n/2} & \text{otherwise} \end{cases} \qquad\qquad N_u = \begin{cases} 0 & \text{if } 2u\geq n \\ \binom{n}{u} & \text{if } 2u<n \end{cases} \quad \text{for } u\geq 3$$

Proof. Theorem 2.3.1 follows from Theorem 2.2.6 by grouping equilibria by their indices. We will consider cases based on n.

- $n = 2$

  The stable equilibrium $\phi = (1,1)$ is the only element of $\Phi_{K,\text{unbalanced}}$. Also $\phi = (1,-1)$ is the only element of $\Phi_{K,\text{balanced}}$. Therefore, $N_0 = 1$ and $N_1 = 1$.

- $n = 3$

  $\Phi_{K,\text{unbalanced}}$ has the stable equilibrium $(1,1,1)$ and the $\binom{3}{1}$ index one equilibria of $(1,1,-1)$ and its permutations. There is only one way to balance the oscillators up to index permutations, namely the evenly spaced arrangement $\phi = (1,\ e^{i\frac{2\pi}{3}},\ e^{i\frac{4\pi}{3}})$ which gives two positive eigenvalues. Since $\phi_1 = 1$ in order to ignore the shift invariance, we have $\binom{2}{1}$ index two equilibria in $\Phi_{K,\text{balanced}}$.

- $n \geq 4$

  $\Phi_{K,\text{unbalanced}}$ contains $\binom{n}{u}$ equilibria having index u where $u < \frac{n}{2}$. Also, there are infinitely many equilibria in $\Phi_{K,\text{balanced}}$ that have index two. For example, let ζ_n be the nth root of unity. Then the last two oscillators can be anywhere on $\mathbb{U}$ as long as they are opposite each other,

  $$\phi = \left(1, \zeta_{n-2}, \zeta_{n-2}^2, \ldots, \zeta_{n-2}^{n-3}, \alpha, -\alpha\right)$$

  Note also that balanced arrangements exist for which no subset is an evenly spaced arrangement, such as the fourth example $\phi^{(4)}$ from $\Phi_{K,\text{balanced}}$ in Example 2.2.8. If n is even, then there are $\binom{n-1}{n/2}$ equilibria having index one in $\Phi_{K,\text{balanced}}$

Combining all these options gives the result. ■

Since there are infinitely many equilibria of index two even modulo shift, it is helpful to also consider the number of connected components when all the equilibria are plotted. We will consider four cases based on the value of n.

$n = 2$

When $n = 2$, there are two equilibria modulo shift. $\theta = (0,0)$ with the eigenvalues $\lambda = (-K, 0)$ and $\theta = (0,\pi)$ with the eigenvalues $\lambda = (0, K)$. Graphing these two equilibria with there shift values gives us two connected components when viewed modulo 2π. A nice way to visualize them in this case is to graph the two equilibria with their shifts on a torus. Figure 2.2 shows these two representations and it is clear that there are two connected components.

$n = 3$

When $n = 3$, there are six equilibria modulo shift. The only stable equilibria is $\theta = (0,0,0)$ and its shifts. Three of the equilibria have index one, namely $(0,0,\pi), (0,\pi,0)$, and $(\pi,0,0)$ and their shifts.

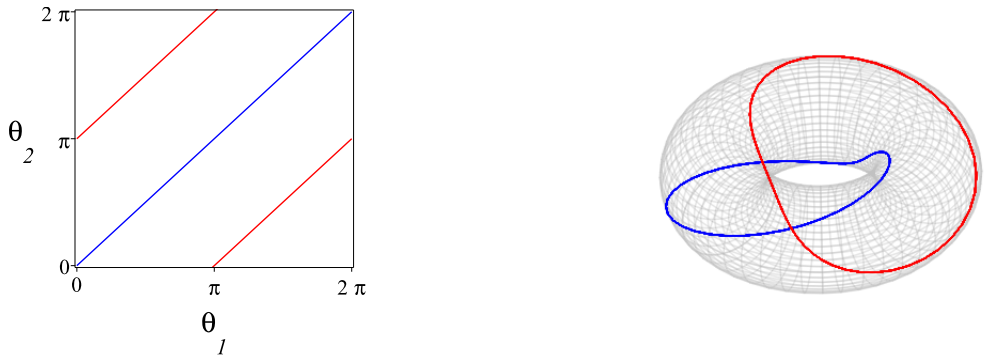


Figure 2.2 Two representations of the connected components of the equilibria of Eq. 2.0.1 when $n = 2$ graphed modulo 2π. The blue line contains all the stable equilibria and the red line contains all the unstable equilibria which have index one.

The fifth and sixth equilibria are balanced equilibria with index two and have the forms $(0, \frac{2\pi}{3}, \frac{4\pi}{3})$ and $(0, \frac{4\pi}{3}, \frac{2\pi}{3})$ and their shifts. Each of these equilibria with their shifts forms a distinct line as can be seen in Figure 2.3 giving six connected components.

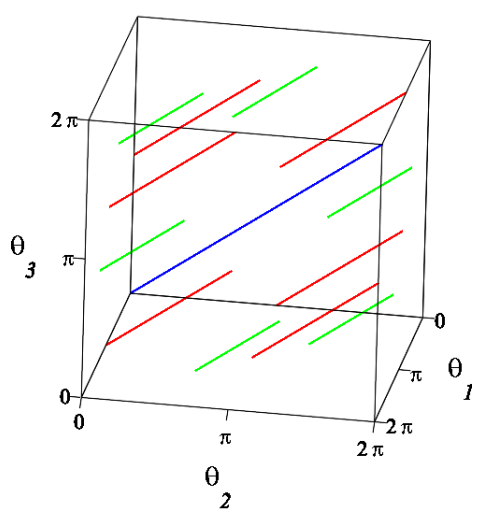


Figure 2.3 The connected components of the equilibria of Eq. 2.0.1 when $n = 3$ graphed modulo 2π. The blue curve contains all the stable equilibria, the three red curves contain all the unstable equilibria of index one, and the two green curves contains all the unstable equilibria of index two.

$n = 4$

When $n = 4$, $f^{-1}(\Phi_{K,\text{unbalanced}})$ contains one stable equilibria, namely $(0,0,0,0)$, and four unstable equilibria of index one, namely $(0,0,0,\pi)$ and its four permutations. However, $f^{-1}(\Phi_{K,\text{balanced}})$ contains an infinite number of equilibria. It has three more index one equilibria, namely $(0,0,\pi,\pi)$, $(0,\pi,0,\pi)$, and $(0,\pi,\pi,0)$, but infinitely many index two equilibria. The following Lemma shows that all the equilibria from $\Phi_{K,\text{balanced}}$ have the same form.

Lemma 2.3.2 *$\phi \in \Phi_{K,balanced}$ when $n = 4$, if and only if ϕ has the form up to shift and index permutation*

$$\phi = (1,-1,\alpha,-\alpha)$$

for some $\alpha \in \mathbb{U}$.

Proof. Let $n = 4$. For an arbitrary $\phi \in \Phi_{K,\text{balanced}}$ we have $\sum_{u=1}^{n} \phi_u = 0$ and $\phi_1 = 1$. Since we are free to rotate all the oscillators simultaneously by a fixed amount and reindex, we can without loss of generality assume that $\phi_1 = \beta$ and $\phi_2 = \beta^*$ for some $\beta \in \mathbb{U}$ with $\text{Re}(\beta), \text{Im}(\beta) \geq 0$. Here we consider two cases.

- $\phi_1 = i$:

  In this case $\phi_2 = -i$, so we must have $\phi_3 + \phi_4 = 0$ to be balanced. Hence $\phi = (i,-i,\gamma,-\gamma)$ for some $\gamma \in \mathbb{U}$. Rotating clockwise by $\frac{\pi}{2}$ then gives the equivalent equilibria $\phi = (1,-1,\alpha,-\alpha)$ for some $\alpha \in \mathbb{U}$.

- $\phi_1 \neq i$:

  In this case we must have that $\text{Re}(\phi_1 + \phi_2) = -\text{Re}(\phi_3 + \phi_4)$ and $\text{Im}(\phi_3) = -\text{Im}(\phi_4)$ for ϕ to be balanced. Since $\text{Re}(\phi_1) = \text{Re}(\phi_2) > 0$, we must have that $\phi_3 = \phi_4^*$ with $\text{Re}(\phi_3) = \text{Re}(\phi_4) = -\text{Re}(\phi_1)$. Thus we have that $\phi = (\beta,\gamma,-\beta,-\gamma)$ for some $\beta,\gamma \in \mathbb{U}$. Rotating so that $\phi_1 = 1$ and reindexing by switching index two with three gives the form $\phi = (1,-1,\alpha,-\alpha)$ for some $\alpha \in \mathbb{U}$.

Thus any $\phi \in \Phi_{K,\text{balanced}}$ has the stated form. Furthermore, it is obvious that $\phi = (1,-1,\alpha,-\alpha) \in \Phi_{K,\text{balanced}}$. ■

Using this Lemma, we can now show that balanced equilibria are one connected component.

Proposition 2.3.3 *The equilibria from $\Phi_{K,balanced}$ when $n = 4$ and when considered with shifts form one connected component.*

Proof. Consider two arbitrary equilibria from $\Phi_{K,\text{balanced}}$,

$$\phi = (1,-1,\alpha,-\alpha) \qquad\qquad \phi' = (1,\beta,-1,-\beta)$$

for some $\alpha,\beta \in \mathbb{U}$. We will show that a continuous path of balanced equilibria exist which connect ϕ and ϕ'. Starting at ϕ, we can continuously rotate α and $-\alpha$ simultaneously until we reach the

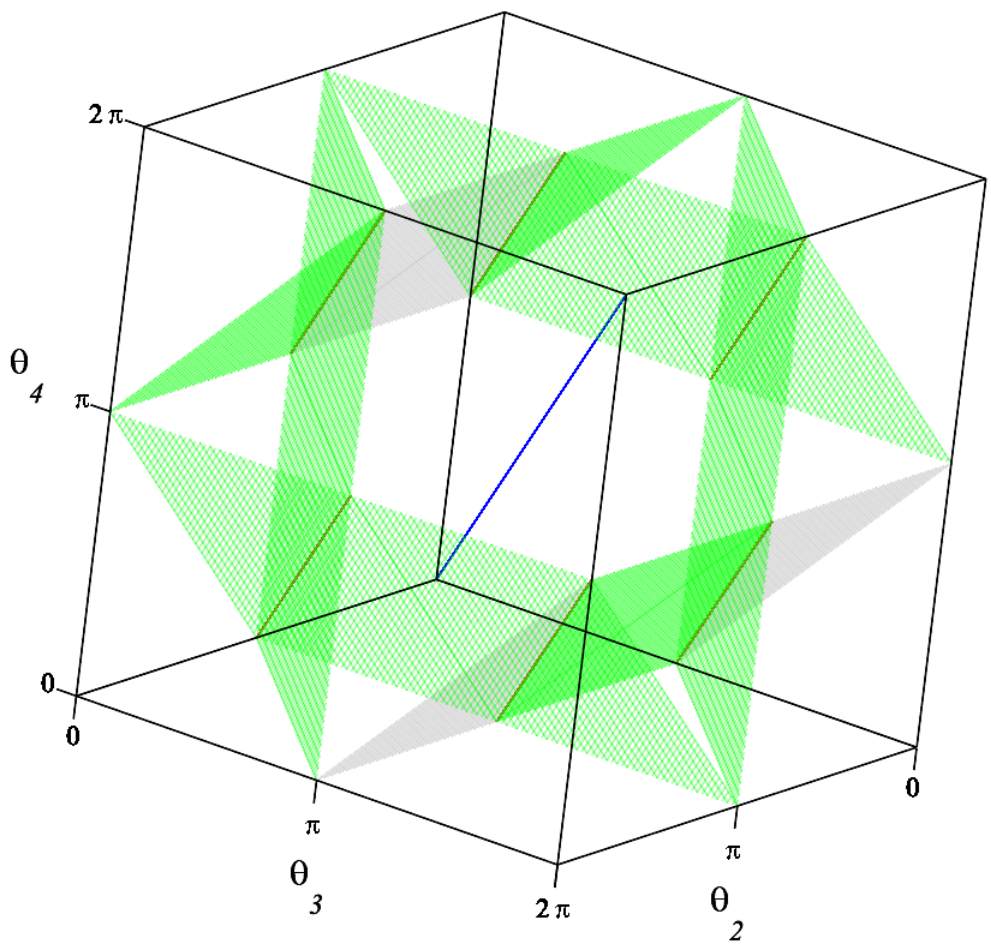


Figure 2.4 The connected components of the equilibria of Eq. 2.0.1 when $n = 4$ projected into $\theta_1 = 0$ space and graphed modulo 2π. The blue line contains all the stable equilibria, the three red lines (at the intersections of the planes) contain all the balanced equilibria of index one, and the green and gray planes contain all the balanced equilibria of index two. The gray plane is all the equilibria with the form modulo shift $(1, -1, \alpha, -\alpha)$, and the green planes contain the other five permutations. Note that the four lines of index one solutions that are unbalanced project onto the same lines as the three index one balanced equilibria and the stable equilibria, but are distinct in $(-\pi, \pi]^4$.

equilibria $(1,-1,-1,1)$. By Lemma 2.3.2, every element along this path is in $\Phi_{K,\text{balanced}}$. Next, we can continuously rotate the second and fourth elements simultaneously until we reach ϕ'. Again by Lemma 2.3.2, every element along this path is in $\Phi_{K,\text{balanced}}$. Therefore ϕ and ϕ' are part of the same connected component.

Repeating this process for any permutation of oscillators shows that all of $\Phi_{K,\text{balanced}}$ is one connected component. ■

Using this result, we have that the equilibria in the $n=4$ case form six connected components. The stable equilibria and index one unbalanced equilibria each form one line for a total of five components, and the balanced equilibria form one higher dimensional connected component. Contained within this component are three lines that are index one, while everywhere else is index two. Figure 2.4 shows the connected components when projected into $\theta_1 = 0$ space.

$n \geq 5$

Using Theorem 2.2.6, it is clear that all of the unbalanced equilibria each form a distinct line in n dimensional space. Thus we have $\binom{n}{u}$ connected components of index u for each $u = 0, 1, \ldots, \left\lfloor \frac{n-1}{2} \right\rfloor$. Furthermore, we conjecture that $\Phi_{K,\text{balanced}}$ behaves the same as it does in the $n=4$ case.

Conjecture 2.3.4 *The equilibria from $\Phi_{K,balanced}$ when $n \geq 5$ and when considered with shifts form one connected component.*

As the number of oscillators increases, it becomes more difficult to classify all the forms of balanced equilibria. However the following example for $n=5$ shows that two groups of balanced oscillators are connected within the same component.

Example 2.3.5 *Let $\phi = (1, \zeta_5, \zeta_5^2, \zeta_5^3, \zeta_5^4)$ and let $\phi' = (1, i, \zeta_3, \zeta_3^2, -i)$ where ζ_n is the nth root of unity. In other words, ϕ is the balanced equilibria where the five oscillators are spaced equidistant along the unit circle, and ϕ' is the composition of two balanced arrangements of two and three oscillators. Starting at ϕ, we can shift the third and fourth oscillators toward each other by the same amount while also shifting the second and fifth oscillators toward the imaginary axis a corresponding amount to keep the arrangement balanced until ϕ' is reached.*

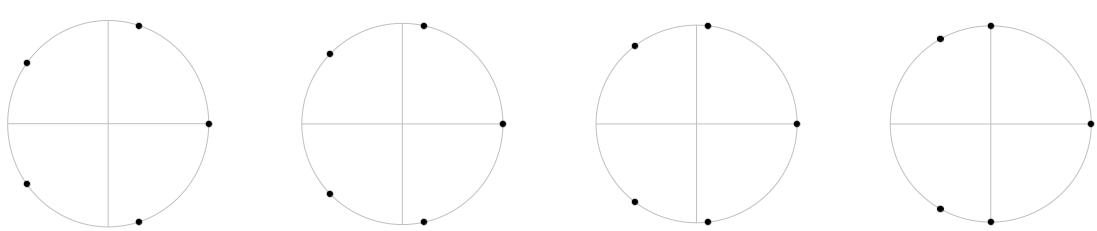


Figure 2.5 The leftmost graph is of ϕ and the middle two graphs show steps along the path to ϕ' which is the rightmost graph.

CHAPTER

3

RANK ONE COUPLING WITH IDENTICAL OSCILLATORS

The Kuramoto model with rank one coupling has the form

$$\frac{\mathrm{d}\theta_u}{\mathrm{d}t} = \omega_u - \frac{1}{n}\sum_{v=1}^{n} k_u k_v \sin(\theta_u - \theta_v) \qquad \text{for } u = 1, 2, \ldots, n \tag{3.0.1}$$

where $n \geq 2$ is the number of oscillators, $k_u > 0$ determines the coupling strength, θ is the phase angles of the oscillators, and ω is the natural frequencies of the oscillators. When all of the oscillators are identical, we have $\omega_1 = \omega_2 = \ldots = \omega_n$. Using the rotating reference frame explained in Section 1.1.1, we can take $\omega_u = 0$ and $\frac{\mathrm{d}\theta_u}{\mathrm{d}t} = 0$ for all u without loss of generality. Thus the equations for the equilibria simplify to

$$0 = \frac{1}{n}\sum_{v=1}^{n} k_u k_v \sin(\theta_u - \theta_v) \qquad \text{for } u = 1, 2, \ldots, n \tag{3.0.2}$$

3.1 Finding

Note that if θ is a solution to Eq. 3.0.1, then $\theta + c$ is also a solution for any constant c. Thus, two solutions are called equivalent modulo shift if each component-wise difference is the same modulo 2π. In order to choose a representative, an output condition for $\theta \in (-\pi, \pi]^n$ is given below so that only one of these equivalent solutions is chosen.

OC: $\quad \sum_{u=1}^{n} k_u e^{i\theta_u} \in \mathbb{R}_+$ or $\left(\sum_{u=1}^{n} k_u e^{i\theta_u} = 0 \text{ and } \theta_1 = 0\right)$.

This choice of representative has a physical interpretation. If weights with masses given by k are placed on the unit circle in $\mathbb{C}$ at angles given by θ, then this choice of representative is the rotation where the center of mass lies on the positive real axis. This choice is always possible and unique provided that the center of mass is not exactly on the origin. In that situation, instead the first oscillator will be fixed to have a phase angle of zero.

Definition 3.1.1 *Let Θ_k be the set of all solutions to Eq. 3.0.2 satisfying* **OC***, that is,*

$$\Theta_k := \left\{ \theta \in (-\pi, \pi]^n : \underset{u}{\forall}\ 0 = \frac{1}{n} \sum_{v=1}^{n} k_u k_v \sin(\theta_u - \theta_v) \wedge \mathbf{OC} \right\}$$

Note that $\mathbb{U}$ is the unit circle in $\mathbb{C}$. In other words, $\mathbb{U} := \{z \in \mathbb{C} \colon |z| = 1\}$. We can perform a convenient transformation on Θ_k and break it into two disjoint sets that characterize the equilibria.

Theorem 3.1.2 *Let*

$$f : \left([\theta_1, \ldots, \theta_n]^{\mathrm{T}}\right) \longmapsto \left[e^{i\theta_1}, \ldots, e^{i\theta_n}\right]^{\mathrm{T}}.$$

Then we have

1. *$f|_{\Theta_k}$ is injective.*
2. *$f(\Theta_k) = \Phi_{k,unbalanced} \uplus \Phi_{k,balanced}$ where*

$$\Phi_{k,unbalanced} := \left\{ \phi \in \{-1, +1\}^n : \sum_{u=1}^{n} k_u \phi_u > 0 \right\}$$
$$\Phi_{k,balanced} := \left\{ \phi \in \mathbb{U}^n : \sum_{u=1}^{n} k_u \phi_u = 0 \wedge \phi_1 = 1 \right\}$$

Proof.

1. Let $\theta, \theta' \in \Theta_k$ be such that $f(\theta) = f(\theta')$. Then for every u we have $e^{i\theta_u} = e^{i\theta'_u}$, and thus $\theta_u = \theta'_u$ since $\theta_u, \theta'_u \in (-\pi, \pi]$. Hence $\theta = \theta'$.
2. Let $r := \frac{1}{n} \sum_{u=1}^{n} k_u \phi_u$ where $\phi_u := e^{i\theta_u}$. Note that z^* denotes the complex conjugate of z. Then we have that

$$\begin{aligned} f(\Theta_k) &= \left\{ f(\theta) : \underset{u}{\forall}\ 0 = \frac{1}{n} \sum_{v=1}^{n} k_u k_v \sin(\theta_u - \theta_v) \wedge \mathbf{OC} \right\} \\ &= \left\{ f(\theta) : \underset{u}{\forall}\ 0 = \frac{1}{n} \sum_{v=1}^{n} k_u k_v \operatorname{Im}\left(e^{i(\theta_u - \theta_v)}\right) \wedge \mathbf{OC} \right\} \\ &= \left\{ \phi \in \mathbb{U}^n : \underset{u}{\forall}\ 0 = \frac{1}{n} \sum_{v=1}^{n} k_u k_v \operatorname{Im}\left(\phi_u \phi_v^*\right) \wedge \left(r \in \mathbb{R}_+ \vee (r = 0 \wedge \phi_1 = 1)\right) \right\} \end{aligned}$$

$$
\begin{aligned}
&= \left\{ \phi \in \mathbb{U}^n : \underset{u}{\forall}\ 0 = \operatorname{Im}\left(\frac{1}{n} \sum_{\nu=1}^{n} k_u k_\nu \phi_u \phi_\nu^* \right) \wedge \left(r \in \mathbb{R}_+ \vee (r = 0 \wedge \phi_1 = 1) \right) \right\} \\
&= \left\{ \phi \in \mathbb{U}^n : \underset{u}{\forall}\ 0 = \operatorname{Im}\left(k_u \phi_u \frac{1}{n} \sum_{\nu=1}^{n} k_\nu \phi_\nu^* \right) \wedge \left(r \in \mathbb{R}_+ \vee (r = 0 \wedge \phi_1 = 1) \right) \right\} \\
&= \left\{ \phi \in \mathbb{U}^n : \underset{u}{\forall}\ 0 = \operatorname{Im}\left(k_u \phi_u r^* \right) \wedge \left(r \in \mathbb{R}_+ \vee (r = 0 \wedge \phi_1 = 1) \right) \right\} \\
&= \left\{ \phi \in \mathbb{U}^n : \underset{u}{\forall}\ 0 = r \operatorname{Im}\left(\phi_u \right) \wedge \left(r \in \mathbb{R}_+ \vee (r = 0 \wedge \phi_1 = 1) \right) \right\} \\
&= \left\{ \phi \in \mathbb{U}^n : \underset{u}{\forall}\ 0 = \operatorname{Im}(\phi_u) \wedge r \in \mathbb{R}_+ \right\} \uplus \left\{ \phi \in \mathbb{U}^n : r = 0 \wedge \phi_1 = 1 \right\} \\
&= \Phi_{k,\text{unbalanced}} \uplus \Phi_{k,\text{balanced}}
\end{aligned}
$$

■

Corollary 3.1.3 *For $\phi \in \Phi_{k,unbalanced} \uplus \Phi_{k,balanced}$, we have that*

$$
f^{-1}(\phi) = \left[\arg(\phi_1), \arg(\phi_2), \ldots, \arg(\phi_n) \right]^{\mathrm{T}} \in \Theta_k
$$

3.2 Stability

In this section we will examine all of the eigenvalues of the equilibria and show that there is one stable equilbirium (in the orbitally stable sense; see Section 1.1.2 for a discussion of orbital stability). Let J be the Jacobian for Eq. 3.0.1. A straightforward calculation gives

$$
J = -\frac{1}{n} \begin{bmatrix} \sum_{\nu=1}^{n} k_1 k_\nu \cos(\theta_1 - \theta_\nu) & & \\ & \ddots & \\ & & \sum_{\nu=1}^{n} k_n k_\nu \cos(\theta_n - \theta_\nu) \end{bmatrix} + \frac{1}{n} \Big[k_u k_\nu \cos(\theta_u - \theta_\nu) \Big]_{(u,\nu)} \tag{3.2.1}
$$

Lemma 3.2.1 *The Jacobian matrix has a zero eigenvalue.*

Proof. It is immediate from Eq. 3.2.1 that the sum of all the columns of J is zero. Therefore the all ones vector is an eigenvector with the corresponding eigenvalue of zero. ■

The all ones eigenvector corresponds to the invariance under shift of the Kuramoto model. Adding the same amount to every phase angle does not change the differences, so we will ignore it when defining stability for the Kuramoto model. Also note that since J is a real symmetric matrix, all the eigenvalues are real. Thus if J evaluated at an equilibrium has $n-1$ negative eigenvalues, then that equilibrium is stable. Likewise, if J has a positive eigenvalue, then that equilibrium cannot be stable [9].

Remark 3.2.2 *For convenience, we will refer to $\phi \in \Phi_{k,unbalanced} \uplus \Phi_{k,balanced}$ as stable if the corresponding $\theta = f^{-1}(\phi)$ is stable.*

It is convenient for the proof later to convert the Jacobian into the same variables as the previous section used.

Proposition 3.2.3 *Let*

$$\phi_u := e^{i\theta_u} \qquad\qquad r := \frac{1}{n}\sum_{u=1}^{n}\phi_u$$

Then for any $\phi \in \Phi_{k,unbalanced} \uplus \Phi_{k,balanced}$*, we have that* $J = D + MM^{\mathrm{T}}$ *where*

$$D := -\begin{bmatrix} k_1 r\,\mathrm{Re}(\phi_1) & & \\ & \ddots & \\ & & k_n r\,\mathrm{Re}(\phi_n) \end{bmatrix} \qquad M := \frac{1}{\sqrt{n}}\begin{bmatrix} k_1\,\mathrm{Re}(\phi_1) & k_1\,\mathrm{Im}(\phi_1) \\ \vdots & \vdots \\ k_n\,\mathrm{Re}(\phi_n) & k_n\,\mathrm{Im}(\phi_n) \end{bmatrix}$$

Proof. Applying the change of variables to Eq. 3.2.1 gives

$$J = -\begin{bmatrix} k_1\,\mathrm{Re}(\phi_1 r^\star) & & \\ & \ddots & \\ & & k_n\,\mathrm{Re}(\phi_n r^\star) \end{bmatrix} + \frac{1}{n}\Big[k_u k_v\,\mathrm{Re}(\phi_u \phi_v^\star)\Big]_{(u,v)}$$

Using the fact that **OC** implies $r \in \mathbb{R}_{\geq 0}$ for an equilibrium and factoring the second matrix gives

$$\begin{aligned} J =& -\begin{bmatrix} k_1 r\,\mathrm{Re}(\phi_1) & & \\ & \ddots & \\ & & k_n r\,\mathrm{Re}(\phi_n) \end{bmatrix} \\ &+ \frac{1}{n}\begin{bmatrix} k_1\,\mathrm{Re}(\phi_1) & k_1\,\mathrm{Im}(\phi_1) \\ \vdots & \vdots \\ k_n\,\mathrm{Re}(\phi_n) & k_n\,\mathrm{Im}(\phi_n) \end{bmatrix}\begin{bmatrix} k_1\,\mathrm{Re}(\phi_1) & \cdots & k_n\,\mathrm{Re}(\phi_n) \\ k_1\,\mathrm{Im}(\phi_1) & \cdots & k_n\,\mathrm{Im}(\phi_n) \end{bmatrix} \\ =& D + MM^{\mathrm{T}} \end{aligned}$$

■

Theorem 3.2.4 $\theta = (0, 0, \ldots, 0)$ *is a stable equilibrium of Eq. 3.0.1. Furthermore, it is the only stable equilibrium.*

Proof. Consider the two sets of equilibria from Theorem 3.1.2.

- $\phi \in \Phi_{k,\text{balanced}}$

  We have that $r = 0$ for any balanced equilibrium. Thus $J = MM^{\mathrm{T}} \succeq 0$ with at least one positive eigenvalue, which means every balanced equilibria is unstable.

- $\phi \in \Phi_{k,\text{unbalanced}}$

By Lemma 1.2.1 and Proposition 3.2.3, we have that

$$\lambda_u(J) = \lambda_u(D + MM^{\mathrm{T}}) \geq \lambda_u(D)$$

since $MM^{\mathrm{T}} \succeq 0$. Therefore, any unbalanced equilibrium with $\phi_u = -1$ for any u is unstable. The only case left to consider is $\phi = (1, \ldots, 1)$. Plugging this into the Jacobian using Proposition 3.2.3 gives

$$J = -\begin{bmatrix} k_1 r & & \\ & \ddots & \\ & & k_n r \end{bmatrix} + \frac{1}{n} k k^{\mathrm{T}}$$

where $r = \frac{1}{n}\sum_{u=1}^{n} k_u$. Again by Lemma 1.2.1 we have that

$$\lambda_u(-J) = \lambda_u\left(-D - \frac{1}{n} k k^{\mathrm{T}}\right) > \lambda_u\left(-\frac{1}{n} k k^{\mathrm{T}}\right)$$

since $-D \succ 0$ when $\phi = (1, \ldots, 1)$. Flipping the sign gives

$$\lambda_u(J) < \lambda_u\left(\frac{1}{n} k k^{\mathrm{T}}\right)$$

since $\frac{1}{n} k k^{\mathrm{T}}$ is a rank one matrix, it has $n-1$ copies of 0 for its eigenvalues. Therefore J must have $n-1$ negative eigenvalues. Therefore $\phi = (1, 1 \ldots, 1)$ is stable.

■

CHAPTER

4

RANK ONE COUPLING WITH NONIDENTICAL OSCILLATORS

The Kuramoto model with rank one coupling has the form

$$\frac{\mathrm{d}\theta_u}{\mathrm{d}t} = \omega_u - \frac{1}{n}\sum_{v=1}^{n} k_u k_v \sin(\theta_u - \theta_v) \qquad \text{for } \; u = 1, 2, \ldots, n \tag{4.0.1}$$

where $n \geq 2$ is the number of oscillators, $k_u > 0$ determines the coupling strength, θ is the phase angles of the oscillators, and ω is the natural frequencies of the oscillators. By using the rotating reference frame explained in Section 1.1.1, the equilibria can be found by solving the set of equations

$$0 = \omega_u - \frac{1}{n}\sum_{v=1}^{n} k_u k_v \sin(\theta_u - \theta_v) \qquad \text{for } \; u = 1, 2, \ldots, n \tag{4.0.2}$$

provided we assume the following input condition.

IC1: $\sum_{u=1}^{n} \omega_u = 0$

Additionally, this chapter will be focused on the case where the oscillators are not identical as the previous chapter covers the identical oscillator case. This gives a second input condition.

IC2: $\omega \neq \vec{0}$

Finally, we will reindex the oscillators by the size of $|\omega_u / k_u|$. This will be convenient in the optimization section.

IC3: $\left|\frac{\omega_1}{k_1}\right| \leq \left|\frac{\omega_2}{k_2}\right| \leq \cdots \leq \left|\frac{\omega_n}{k_n}\right|$

Remark 4.0.1 *Some of the results covered in this chapter were developed in collaboration with Jonathan Hauenstein, Hoon Hong, and Daniel Molzahn and were published in [15].*

4.1 Finding

Note that if θ is a solution of Eq. 4.0.1 then any shift of θ (modulo 2π) is also an equilibrium. Thus we will impose a condition on θ so that only one of these "equivalent" solutions is chosen.

OC: $$\sum_{u=1}^{n} k_u e^{i\theta_u} \in \mathbb{R}_+$$

This choice of representative has a physical interpretation. If weights with masses given by k are placed on the unit circle in $\mathbb{C}$ at angles given by θ, then this choice of representative is the rotation where the center of mass lies on the positive real axis. This choice is always possible and unique provided that the center of mass is not exactly on the origin. Lemma 4.1.2 shows that this edge case is avoided whenever **IC2** is satisfied.

Definition 4.1.1 *Let $\Theta_{k,\omega}$ be the set of all solutions to Eq. 4.0.1 satisfying* **OC***, that is,*

$$\Theta_{k,\omega} := \left\{ \theta \in (-\pi, \pi]^n : \underset{u}{\forall}\ \omega_u = \frac{1}{n}\sum_{v=1}^{n} k_u k_v \sin(\theta_u - \theta_v) \wedge \mathbf{OC} \right\}$$

Note that $\mathbb{U}$ is the unit circle in $\mathbb{C}$. In other words, $\mathbb{U} := \{z \in \mathbb{C}\colon\ |z| = 1\}$. We can perform a series of two transformation on $\Theta_{k,\omega}$ to decouple the equations and instead have univariate radical equations.

Lemma 4.1.2 *Let*

$$f : \left([\theta_1, \ldots, \theta_n]^{\mathrm{T}}\right) \longmapsto \left[e^{i\theta_1}, \ldots, e^{i\theta_n}\right]^{\mathrm{T}}.$$

Then we have

1. *$f|_{\Theta_{k,\omega}}$ is injective.*
2. *$f\left(\Theta_{k,\omega}\right) = \Phi_{k,\omega}$ where*

$$\Phi_{k,\omega} := \left\{ \phi \in \mathbb{U}^n : \underset{u}{\forall}\ \omega_u = k_u r \operatorname{Im}\left(\phi_u\right) \wedge r \in \mathbb{R}_+ \right\}$$

and where $r := \frac{1}{n}\sum_{u=1}^{n} k_u \phi_u$.

Remark 4.1.3 *Note that if $r = 0$, then we must have that $\omega = \vec{0}$, which contradicts **IC2**. Therefore **IC2** implies $r \neq 0$.*

Proof.

1. Let $\theta, \theta' \in \Theta_{k,\omega}$ be such that $f(\theta) = f(\theta')$. Then for every u we have $e^{i\theta_u} = e^{i\theta'_u}$, and thus $\theta_u = \theta'_u$ since $\theta_u, \theta'_u \in (-\pi, \pi]$. Hence $\theta = \theta'$.

2. Let $r := \frac{1}{n}\sum_{u=1}^{n} k_u \phi_u$ where $\phi_u := e^{i\theta_u}$. Note that z^* denotes the complex conjugate of z. Then we have that

$$\begin{aligned}
f\left(\Theta_{k,\omega}\right) &= \left\{ f(\theta) : \underset{u}{\forall}\ \omega_u = \frac{1}{n}\sum_{v=1}^{n} k_u k_v \sin(\theta_u - \theta_v) \wedge \mathbf{OC} \right\} \\
&= \left\{ f(\theta) : \underset{u}{\forall}\ \omega_u = \frac{1}{n}\sum_{v=1}^{n} k_u k_v \operatorname{Im}\left(e^{i(\theta_u - \theta_v)}\right) \wedge \mathbf{OC} \right\} \\
&= \left\{ \phi \in \mathbb{U}^n : \underset{u}{\forall}\ \omega_u = \frac{1}{n}\sum_{v=1}^{n} k_u k_v \operatorname{Im}\left(\phi_u \phi_v^*\right) \wedge r \in \mathbb{R}_+ \right\} \\
&= \left\{ \phi \in \mathbb{U}^n : \underset{u}{\forall}\ \omega_u = \operatorname{Im}\left(\frac{1}{n}\sum_{v=1}^{n} k_u k_v \phi_u \phi_v^* \right) \wedge r \in \mathbb{R}_+ \right\} \\
&= \left\{ \phi \in \mathbb{U}^n : \underset{u}{\forall}\ \omega_u = \operatorname{Im}\left(k_u \phi_u \frac{1}{n}\sum_{v=1}^{n} k_v \phi_v^* \right) \wedge r \in \mathbb{R}_+ \right\} \\
&= \left\{ \phi \in \mathbb{U}^n : \underset{u}{\forall}\ \omega_u = \operatorname{Im}\left(k_u \phi_u r^* \right) \wedge r \in \mathbb{R}_+ \right\} \\
&= \left\{ \phi \in \mathbb{U}^n : \underset{u}{\forall}\ \omega_u = k_u r \operatorname{Im}\left(\phi_u \right) \wedge r \in \mathbb{R}_+ \right\}
\end{aligned}$$

■

Lemma 4.1.4 *We have*

$$\Phi_{k,\omega} = \bigcup_{\sigma \in \{-,+\}^n} \Phi_{k,\omega,\sigma}$$

where

$$\Phi_{k,\omega,\sigma} := \left\{ \phi \in \mathbb{U}^n : \underset{u}{\forall}\ \phi_u = \frac{\omega_u i + \sigma_u \sqrt{k_u^2 r^2 - \omega_u^2}}{k_u r} \wedge r \in \mathbb{R}_+ \right\}$$

Proof. We have that

$$\begin{aligned}
\Phi_{k,\omega} &= \left\{ \phi \in \mathbb{U}^n : \underset{u}{\forall}\ \omega_u = k_u r \operatorname{Im}(\phi_u) \wedge r \in \mathbb{R}_+ \right\} \\
&= \left\{ \phi \in \mathbb{U}^n : \underset{u}{\forall}\ 2\omega_u i = k_u r(\phi_u - \phi_u^*) \wedge r \in \mathbb{R}_+ \right\}
\end{aligned}$$

Note that $|\phi_u| = 1$, so $\phi_u \neq 0$. Thus multiplying through by ϕ_u gives us that

$$\begin{aligned}
\Phi_{k,\omega} &= \left\{ \phi \in \mathbb{U}^n : \underset{u}{\forall}\ 2\omega_u \phi_u i = k_u r(\phi_u^2 - 1) \wedge r \in \mathbb{R}_+ \right\} \\
&= \left\{ \phi \in \mathbb{U}^n : \underset{u}{\forall}\ 0 = k_u r \phi_u^2 - 2\omega_u i \phi_u - k_u r \wedge r \in \mathbb{R}_+ \right\}
\end{aligned}$$

Since $k_u r \neq 0$, we can use the quadratic formula to solve for ϕ_u.

$$\begin{aligned}\Phi_{k,\omega} &= \bigcup_{\sigma\in\{-,+\}^n} \left\{\phi \in \mathbb{U}^n : \underset{u}{\forall}\ \phi_u = \frac{2\omega_u i + \sigma_u\sqrt{-4\omega_u^2 + 4k_u^2 r^2}}{2k_u r} \ \wedge\ r \in \mathbb{R}_+\right\} \\ &= \bigcup_{\sigma\in\{-,+\}^n} \left\{\phi \in \mathbb{U}^n : \underset{u}{\forall}\ \phi_u = \frac{\omega_u i + \sigma_u\sqrt{k_u^2 r^2 - \omega_u^2}}{k_u r} \ \wedge\ r \in \mathbb{R}_+\right\}\end{aligned}$$

■

Lemma 4.1.5 *Let*

$$g : \phi \longmapsto \frac{1}{n} k^{\mathrm{T}} \phi.$$

Then we have that

1. $g|_{\Phi_{k,\omega,\sigma}}$ *is injective.*
2. $g(\Phi_{k,\omega,\sigma}) = R_{k,\omega,\sigma}$ *where*

$$R_{k,\omega,\sigma} := \left\{ r \in \mathbb{R}_+ :\ r^2 = \frac{1}{n}\sum_{u=1}^{n} \sigma_u \sqrt{k_u^2 r^2 - \omega_u^2}\right\}$$

Proof.

1. Let $\phi, \phi' \in \Phi_{k,\omega,\sigma}$ be such that $g(\phi) = g(\phi')$. Then

$$r = g(\phi) = g(\phi') = r'$$

   Thus

$$\phi_u = \frac{\omega_u i + \sigma_u\sqrt{k_u^2 r^2 - \omega_u^2}}{k_u r} = \frac{\omega_u i + \sigma_u\sqrt{k_u^2 (r')^2 - \omega_u^2}}{k_u r'} = \phi'_u$$

2. Note that $g(\phi) = r$. Thus we have that

$$\begin{aligned}g(\Phi_{k,\omega,\sigma}) &= \left\{ g(\phi) : \underset{u}{\forall}\ \phi_u = \frac{\omega_u i + \sigma_u\sqrt{k_u^2 r^2 - \omega_u^2}}{k_u r} \ \wedge\ r \in \mathbb{R}_+\right\} \\ &= \left\{ r \in \mathbb{R}_+ :\ r = \frac{1}{n}\sum_{u=1}^{n} k_u \phi_u \ \wedge\ \underset{u}{\forall}\ \phi_u = \frac{\omega_u i + \sigma_u\sqrt{k_u^2 r^2 - \omega_u^2}}{k_u r}\right\} \\ &= \left\{ r \in \mathbb{R}_+ : r = \frac{1}{n}\sum_{u=1}^{n} k_u \left(\frac{\omega_u i + \sigma_u\sqrt{k_u^2 r^2 - \omega_u^2}}{k_u r}\right) \wedge\ \underset{u}{\forall}\ \phi_u = \frac{\omega_u i + \sigma_u\sqrt{k_u^2 r^2 - \omega_u^2}}{k_u r}\right\} \\ &= \left\{ r \in \mathbb{R}_+ :\ r = \frac{1}{n}\sum_{u=1}^{n} k_u \left(\frac{\omega_u i + \sigma_u\sqrt{k_u^2 r^2 - \omega_u^2}}{k_u r}\right)\right\} \\ &= \left\{ r \in \mathbb{R}_+ :\ r^2 = \frac{1}{n}\sum_{u=1}^{n} \omega_u i + \sigma_u\sqrt{k_u^2 r^2 - \omega_u^2}\right\}\end{aligned}$$

$$= \left\{ r \in \mathbb{R}_+ : \; r^2 = \frac{1}{n} \sum_{u=1}^{n} \sigma_u \sqrt{k_u^2 r^2 - \omega_u^2} \right\}$$

since **IC1** gives $\sum_{u=1}^{n} \omega_u = 0$.

■

Putting the previous three Lemmas together allows for a reformulation of $\Theta_{k,\omega}$.

Theorem 4.1.6 *We have*

$$\Theta_{k,\omega} = F\left(\bigcup_{\sigma \in \{-,+\}^n} G_\sigma \left(R_{k,\omega,\sigma} \right) \right)$$

where

$$F(\phi) := \left[\arg(\phi_1), \ldots, \arg(\phi_n) \right]^{\mathrm{T}}$$
$$G_\sigma(r) := \left[\frac{\omega_1 i + \sigma_1 \sqrt{k_1^2 r^2 - \omega_1^2}}{k_1 r}, \ldots, \frac{\omega_n i + \sigma_n \sqrt{k_n^2 r^2 - \omega_n^2}}{k_n r} \right]^{\mathrm{T}}$$

Furthermore, F and G_σ are injective.

Proof. We split the proof into three parts.

1. $F = f^{-1}$ and is injective.

   Recall that
   $$f\left([\theta_1, \ldots, \theta_n]^{\mathrm{T}} \right) = \left[e^{i\theta_1}, \ldots, e^{i\theta_n} \right]^{\mathrm{T}}.$$
   Therefore,
   $$f^{-1}(\phi) = \left[\arg(\phi_1), \ldots, \arg(\phi_n) \right]^{\mathrm{T}} = F(\phi)$$
   and F is injective.

2. $G_\sigma = g^{-1}|_{\Phi_{k,\omega,\sigma}}$ and is injective.

   Recall that
   $$g(\phi) = \frac{1}{n} k^{\mathrm{T}} \phi.$$
   Let $r = g(\phi)$ where $\phi \in \Phi_{k,\omega,\sigma}$. Then
   $$\phi_u = \frac{\omega_u i + \sigma_u \sqrt{k_u^2 r^2 - \omega_u^2}}{k_u r} = G_\sigma(r)_u.$$
   Furthermore, let $r, r' \in R_{k,\omega,\sigma}$ be such that $G_\sigma(r) = G_\sigma(r')$. Then
   $$r = \frac{1}{n} \sum_{u=1}^{n} k_u \frac{\omega_u i + \sigma_u \sqrt{k_u^2 r^2 - \omega_u^2}}{k_u r}$$

$$
\begin{aligned}
&= \frac{1}{n}\sum_{u=1}^{n} k_u G_\sigma(r)_u \\
&= \frac{1}{n}\sum_{u=1}^{n} k_u G_\sigma(r')_u \\
&= \frac{1}{n}\sum_{u=1}^{n} k_u \frac{\omega_u i + \sigma_u \sqrt{k_u^2 (r')^2 - \omega_u^2}}{k_u r'} \\
&= r'
\end{aligned}
$$

3. $\Theta_{k,\omega} = F\left(\bigcup_{\sigma\in\{-,+\}^n} G_\sigma\left(R_{k,\omega,\sigma}\right)\right)$.

   Combining Lemmas 4.1.2, 4.1.4, and 4.1.5 with parts one and two above gives

$$
\Theta_{k,\omega} = f^{-1}\left(\Phi_{k,\omega}\right) = f^{-1}\left(\bigcup_{\sigma\in\{-,+\}^n} \Phi_{k,\omega,\sigma}\right) = f^{-1}\left(\bigcup_{\sigma\in\{-,+\}^n} g^{-1}\left(R_{k,\omega,\sigma}\right)\right)
$$

■

For each $\sigma \in \{-,+\}^n$, the first step to utilize Theorem 4.1.6 for locating all equilibria is to find the positive solutions to

$$
r^2 = \frac{1}{n}\sum_{u=1}^{n} \sigma_u \sqrt{k_u^2 r^2 - \omega_u^2}
$$

This is equivalent to finding the positive roots of

$$
f_\sigma(R) := -R + \frac{1}{n}\sum_{u=1}^{n} \sigma_u \sqrt{k_u^2 R - \omega_u^2} \tag{4.1.1}
$$

where $R := r^2$. For each positive root R of f_σ, the second step from Theorem 4.1.6 is to compute the equilibria via $F\left(G_\sigma\left(\sqrt{R}\right)\right)$. This process is summarized in the following algorithm. Note that it depends upon a root finding method that returns the set of all positive roots of f_σ, denoted $\mathbf{Solve}_+(f_\sigma)$.

Algorithm 4.1.7

In: $\omega \in \mathbb{R}^n$ *and* $k \in \mathbb{R}^n_{>0}$ *satisfying* ***IC1***, ***IC2***, *and* ***IC3***.

Out: $\Theta_{k,\omega}$, *the set of equilibria satisfying* ***OC1***.

1. $\Theta_{k,\omega} \leftarrow \{\}$
2. *For* $\sigma \in \{-,+\}^n$ *do*
   (a) $f_\sigma \leftarrow -R + \frac{1}{n}\sum_{u=1}^{n} \sigma_u \sqrt{k_u^2 R - \omega_u^2}$

(b) $\mathscr{R} \leftarrow \textbf{Solve}_+\left(f_\sigma\right)$

(c) *For* $R \in \mathscr{R}$ *do*

i. $\theta \leftarrow F\left(G_\sigma\left(\sqrt{R}\right)\right)$

ii. *Add* θ *to* $\Theta_{k,\omega}$

Example 4.1.8 *To illustrate for* $n = 2$*, consider* $\omega = (4,-4)$ *and* $k = (5,2)$*. There are* 4 *sign patterns* σ *to consider:*

- $\sigma = (+,+)$:
  - $f_\sigma(R) = -R + \frac{1}{2}\left(\sqrt{25R-16} + \sqrt{4R-16}\right)$ *has one positive root, namely* $R = 10.25$.
  - *This yields the equilibrium* $\theta = (0.2526, -0.6747)$.
- $\sigma = (+,-)$:
  - $f_\sigma(R) = -R + \frac{1}{2}\left(\sqrt{25R-16} - \sqrt{4R-16}\right)$ *has one positive root, namely* $R = 4.25$.
  - *This yields the equilibrium* $\theta = (0.3985, -1.8158)$.
- $\sigma = (-,+)$:
  - $f_\sigma(R) = -R + \frac{1}{2}\left(-\sqrt{25R-16} + \sqrt{4R-16}\right)$ *has no positive roots.*
- $\sigma = (-,-)$:
  - $f_\sigma(R) = -R + \frac{1}{2}\left(-\sqrt{25R-16} - \sqrt{4R-16}\right)$ *has no positive roots.*

In summary, there are two equilibria satisfying Eq. 4.0.1.

Optimizations

In Algorithm 4.1.7, $\textbf{Solve}_+(f_\sigma)$, which computed all positive roots of f_σ, was called for all 2^n sign patterns. This exponential scaling in the number of oscillators is not any better than the previous approaches discussed in Section 1.2.2. As such, the goal of this section is to prune out sign patterns σ for which f_σ defined in Eq. 4.1.1 has no positive roots. This improvement allows for the optimized algorithm to scale much more efficiently. Throughout this section, we assume $\omega \in \mathbb{R}^n$ and $k \in \mathbb{R}^n_{>0}$ satisfy **IC1** – **IC3** and $\sigma \in \{-,+\}^n$.

Before looking at the sign cases, we can also consider some slight improvements in the calculation of θ given r.

Proposition 4.1.9 *Given* $\sigma \in \{-,+\}^n$ *and* $r \in R_{k,\omega,\sigma}$*, let* $\theta = F(G_\sigma(r))$*. Then*

$$\underset{u}{\forall}\ \theta_u = \arcsin\left(\frac{\omega_u}{k_u r}\right) \quad s.t. \quad \operatorname{sign}(\cos(\theta_u)) = \sigma_u$$

provided $\cos(\theta_u) \neq 0$*. If instead* $\cos(\theta_u) = 0$*, then we have that* $\theta_u = \operatorname{sign}(\omega_u)\dfrac{\pi}{2}$.

Proof. From Theorem 4.1.6 we have that

$$F(G_\sigma(r))_u = \arg\left(\frac{\omega_u i + \sigma_u\sqrt{k_u^2 r^2 - \omega_u^2}}{k_u r}\right)$$

Let $\phi_u = G_\sigma(r) = \dfrac{\omega_u i + \sigma_u\sqrt{k_u^2 r^2 - \omega_u^2}}{k_u r}$. Thus $\phi \in \Phi_{k,\omega,\sigma} \subset \Phi_{k,\omega}$. Therefore, $\mathrm{Im}(\phi_u) = \dfrac{\omega_u}{k_u r}$, so we must have $\mathrm{Re}(\phi_u) = \dfrac{\sigma_u\sqrt{k_u^2 r^2 - \omega_u^2}}{k_u r}$. Since $k_u r > 0$, we have that $\mathrm{sign}(\mathrm{Re}(\phi_u)) = \sigma_u$ provided that $\sqrt{k_u^2 r^2 - \omega_u^2} > 0$. Also, $\phi \in \mathbb{U}^n$, so

$$\theta_u = \arcsin\left(\frac{\omega_u}{k_u r_u |\phi_u|}\right) = \arcsin\left(\frac{\omega_u}{k_u r_u}\right) \quad \text{s.t.} \quad \mathrm{sign}(\cos(\theta_u)) = \sigma_u$$

when $\cos(\theta_u) \neq 0$. If however $\mathrm{Re}(\phi_u) = 0$, then we have that $\phi_u = \dfrac{\omega_u i}{k_u r}$. Since $\phi \in \mathbb{U}$ and ϕ is purely imaginary, we get that $\phi_u = \mathrm{sign}(\omega_u) i$, so $\theta_u = \mathrm{sign}(\omega_u)\dfrac{\pi}{2}$. ■

We can also find an interval that contains all the positive root of f_σ.

Proposition 4.1.10 *If $\sigma_+ := \{u : \sigma_u = +\}$, then every positive root of f_σ is contained in the interval*

$$\left[\left(\frac{\omega_n}{k_n}\right)^2, \left(\frac{1}{n}\sum_{u\in\sigma_+} k_u\right)^2\right].$$

Remark 4.1.11 *Note that this always rules out the all negative sign case since $\left(\dfrac{\omega_n}{k_n}\right)^2 > 0$.*

Proof. Suppose that R is a positive root of f_σ. Since each $\dfrac{\omega_u^2}{k_u^2 R} = \sin^2\theta_u \leq 1$ by Proposition 4.1.9,

$$R \geq \left(\frac{\omega_u}{k_u}\right)^2 \quad \text{for } u = 1, 2, \ldots, n.$$

Hence, **IC3** gives us that $R \geq \left(\dfrac{\omega_n}{k_n}\right)^2$. Moreover,

$$0 \leq \sqrt{k_u^2 R - \omega_u^2} \leq k_u\sqrt{R}.$$

Thus for $\sigma_+ := \{u : \sigma_u = +\}$ and $\sigma_- := \{u : \sigma_u = -\}$, we have

$$\begin{aligned} R &= \frac{1}{n}\sum_{u=1}^{n} \sigma_u \sqrt{k_u^2 R - \omega_u^2} \\ &= \left(\frac{1}{n}\sum_{u\in\sigma_+} \sqrt{k_u^2 R - \omega_u^2}\right) - \left(\frac{1}{n}\sum_{u\in\sigma_-} \sqrt{k_u^2 R - \omega_u^2}\right) \\ &\le \frac{1}{n}\sum_{u\in\sigma_+} k_u \sqrt{R}. \end{aligned}$$

This is equivalent to $R \le \left(\frac{1}{n}\sum_{u\in\sigma_+} k_u\right)^2$. ■

Example 4.1.12 *With the setup from Example 4.1.8, we consider the four cases:*

- $\sigma = (+,+)$:
  - *Proposition 4.1.10 gives the interval* $[4, 12.25]$, *which contains the positive root* $R = 10.25$.
- $\sigma = (+,-)$:
  - *Proposition 4.1.10 gives the interval* $[4, 6.25]$, *which contains the positive root* $R = 4.25$.
- $\sigma = (-,+)$:
  - *no positive roots since Proposition 4.1.10 gives the "interval"* $[4, 1]$.
- $\sigma = (-,-)$:
  - *no positive roots since Proposition 4.1.10 gives the "interval"* $[4, 0]$.

As shown in Example 4.1.12, Proposition 4.1.10 can exclude sign patterns σ for which f_σ has no positive roots. The following Lemma will be useful in developing more ways to exclude sign cases.

Lemma 4.1.13 *f_σ has no positive roots if and only if $f_\sigma < 0$ on* $\left[\left(\frac{\omega_n}{k_n}\right)^2, \infty\right)$.

Proof. Let $I = \left[\left(\frac{\omega_n}{k_n}\right)^2, \infty\right)$. If $R \in I$, then

$$\begin{aligned} f_\sigma(R) &= -R + \frac{1}{n}\sum_{u=1}^{n} \sigma_u \sqrt{k_u^2 R - \omega_u^2} \\ &\le -R + \frac{1}{n}\sum_{u=1}^{n} \sqrt{k_u^2 R - \omega_u^2} \\ &\le -R + \frac{1}{n}\sum_{u=1}^{n} k_u \sqrt{R} \end{aligned}$$

so that

$$\lim_{R\to\infty} f_\sigma(R) = -\infty.$$

Since f_σ is continuous on I, we must have $f_\sigma < 0$ on I when f_σ has no positive roots. Furthermore, the interval $\left[\left(\frac{\omega_n}{k_n}\right)^2, \left(\frac{1}{n}\sum_{u\in\sigma_+} k_u\right)^2\right]$ contains all the positive roots of f_σ by Proposition 4.1.10 and is contained in I. Hence, if $f_\sigma < 0$ on I, then f_σ has no positive roots. ■

If f_σ has no positive roots, the following yields additional cases which also have no positive roots.

Proposition 4.1.14 *Let u be such that $\sigma_u = +$. Let σ' such that $\sigma'_u = -$ and $\sigma'_v = \sigma_v$ for $v \neq u$. If f_σ has no positive roots, then $f_{\sigma'}$ also has no positive roots.*

Proof. Since f_σ has no positive roots, Lemma 4.1.13 shows that $f_\sigma < 0$ on $\left[\left(\frac{\omega_n}{k_n}\right)^2, \infty\right)$. Since $f_{\sigma'} \leq f_\sigma$ on $\left[\left(\frac{\omega_n}{k_n}\right)^2, \infty\right)$, $f_{\sigma'}$ also does not have any positive roots by Lemma 4.1.13. ■

Example 4.1.15 *With the setup from Example 4.1.8, since f_σ for $\sigma = (-,+)$ has no positive roots, $f_{\sigma'}$ also has no positive roots for $\sigma' = (-,-)$.*

A natural way to order the sign patterns σ is to convert them using binary representations of the numbers from 0 to $2^n - 1$ where "0" in binary comes from a $-$ and "1" in binary comes from a $+$. We will denote this using $\iota(\sigma)$. For example, $\sigma = (+,-)$ corresponds to the binary number 10_2, so we can say $\iota(\sigma) = 2$. This allows for an important technique. Because merely tracking all of the sign cases could use prohibitive amounts of memory for large n values, a way to skip sign cases efficiently without tracking them all is needed. With a binary numbering scheme, one needs to track only the current sign number and can skip over sequential sign cases using the result of Proposition 4.1.14.

Definition 4.1.16 *Let $\iota : \{-,+\}^n \to \mathbb{Z}$ where ι maps $-$ to "0" and $+$ to "1" in the n digit binary representation of the output.*

Theorem 4.1.17 *Let $\sigma \in \{-,+\}^n$ be such that f_σ has no positive roots and let p be the index of the last negative in σ or zero if there are no negatives in σ. Let $\sigma' \in \{-,+\}^n$ be such that*

- *$\sigma'_u = \sigma_u$ for $u \leq p$.*
- *$\sigma'_u = -$ for $u > p$.*

Then $f_{\hat\sigma}$ has no positive roots for any $\hat\sigma \in \{-,+\}^n$ where $\iota(\sigma') \leq \iota(\hat\sigma) \leq \iota(\sigma)$.

Proof. Let σ, σ', and p be as described in the theorem statement above. Let $\hat\sigma \in \{-,+\}^n$ be such that $\iota(\sigma') \leq \iota(\hat\sigma) \leq \iota(\sigma)$. Since the bits of $\iota(\sigma)$ and $\iota(\sigma')$ agree up to the p-th position, only the less significant bits to the right of the p-th position of $\iota(\hat\sigma)$ can differ from $\iota(\sigma)$. Furthermore, the

bits past the p-th position of $\iota(\sigma)$ are all ones by selection of p. Therefore, $\iota(\hat{\sigma})$ can only have ones changed to zeros when compared to $\iota(\sigma)$. Thus Proposition 4.1.14 can be repeatedly applied for each such position where the bits differ, so $f_{\hat{\sigma}}$ cannot have any positive roots. ■

In order to use Theorem 4.1.17, we will define a function that gives the next value that may need to be checked.

Definition 4.1.18 *The **pruning function** $\mathrm{P}_\ell : \mathbb{Z}_{>0} \to \mathbb{Z}_{\geq 0}$ for $\ell \in \mathbb{Z}_{>0}$ takes the binary representation of m, zeroes out everything to the right of the ℓth zero from the right (left padding with zeros as needed), and then subtracts one. If that number is negative, then it returns 0, otherwise it returns that number.*

Example 4.1.19 *Note that* $23 = 10111_2$. *Then*

$$\mathrm{P}_1(23) = 01111_2 = 15 \qquad\qquad \mathrm{P}_2(23) = 0$$

Example 4.1.20 *Let* $\sigma = (+,-,+,+) \underset{\iota}{\mapsto} 11$ *and suppose f_σ has no positive roots. Using Theorem 4.1.17, we know that the following sign cases have no roots:*

$$(+,-,+,-) \underset{\iota}{\mapsto} 10$$
$$(+,-,-,+) \underset{\iota}{\mapsto} 9$$
$$(+,-,-,-) \underset{\iota}{\mapsto} 8$$

Note that Proposition 4.1.14 could be used to rule out additional sign cases such as $(-,-,+,+) \underset{\iota}{\mapsto} 3$, *however such sign cases are not sequential. Moreover, it is trivial to compute that* $\mathrm{P}_1(11) = 7$ *gives the next case that may need to be checked by zeroing out the bits to the right of the last zero of $\iota(\sigma)$ and subtracting one.*

Applying Theorem 4.1.17 allows for a significantly improved algorithm. The following algorithm depends upon a root finding method that returns the set of roots of f_σ within the interval I denoted **Solve**(f_σ, I).

Algorithm 4.1.21 (Prune 1)

In: $\omega \in \mathbb{R}^n$ *and* $k \in \mathbb{R}^n_{>0}$ *satisfying* **IC1–IC3**.

Out: $\Theta_{k,\omega}$

1. $\Theta_{k,\omega} \leftarrow \{\}$
2. $j \leftarrow 2^n - 1$
3. *While* $j > 0$ *do*
   (a) $\sigma \leftarrow \iota^{-1}(j)$

(b) $I \leftarrow \left[\left(\frac{\omega_n}{k_n}\right)^2, \left(\frac{1}{n}\sum_{u\in\sigma_+} k_u\right)^2 \right]$

(c) *If I is empty, then*

   i. $j \leftarrow \mathrm{P}_1(u)$

   ii. *Continue (go back to Step 3)*

(d) $f_\sigma \leftarrow -R + \frac{1}{n}\sum_{u=1}^{n} \sigma_u \sqrt{k_u^2 R - \omega_u^2}$

(e) $\mathscr{R} \leftarrow \mathbf{Solve}(f_\sigma, I)$

(f) *If $\mathscr{R} = \emptyset$, then*

   i. $j \leftarrow \mathrm{P}_1(u)$

   ii. *Continue (go back to Step 3)*

(g) *For $R \in \mathscr{R}$ do*

   i. *Compute $\theta \in (-\pi, \pi]^n$ according to Proposition 4.1.9*

   ii. *Add θ to $\Theta_{k,\omega}$*

(h) $j \leftarrow u - 1$

Here we consider the above algorithm on a concrete application (power flow analysis) from electrical engineering [21].

Example 4.1.22 (Power flow model: 4-bus system) *Figure 4.1 depicts a lossless four-bus power system with active power injections p, voltage magnitudes $|V|$, and line susceptances $b_{12} = b_{13} = b_{14} = b_{23} = b_{24} = b_{34} = -1$. The equilibria of the power flow equations correspond to the equilibria of the rank-one coupled Kuramoto model, namely the solutions of* (4.0.1) *where $\omega = (p_1, p_2, p_3, p_4)$, $k = (2|V_1|,\ 2|V_2|,\ 2|V_3|,\ 2|V_4|)$, and $\theta = (\theta_1, \theta_2, \theta_3, \theta_4)$ are the voltage angles.*

Suppose that $p = (1.00, -1.25, 2.00, -1.75)$ and $|V| = (1.10, 0.93, 1.05, 0.90)$. Algorithm 4.1.21 starts with $j = 15$.

- *The case with $\sigma = \iota^{-1}(15)$ produces a positive root, so the algorithm proceeds with $j = 14$. The cases $j = 14, 13, 12, 11, 10$ also produce a positive root, so proceed in turn.*

- *The case with $\sigma = \iota^{-1}(9)$ produces no roots. Using Theorem 4.1.17, case $j = 8$ can be skipped, so $j = 7$ is next considered.*

- *The case with $\sigma = \iota^{-1}(7)$ produces two roots, so $j = 6$ is next considered.*

- *The case with $\sigma = \iota^{-1}(6)$ produces no roots. Theorem 4.1.17 does not give any additional cases that can be skipped in this case, so $j = 5$ is next considered.*

- *The case with $\sigma = \iota^{-1}(5)$ produces no roots. Theorem 4.1.17 allows case $j = 4$ to be skipped, so $j = 3$ is next considered.*

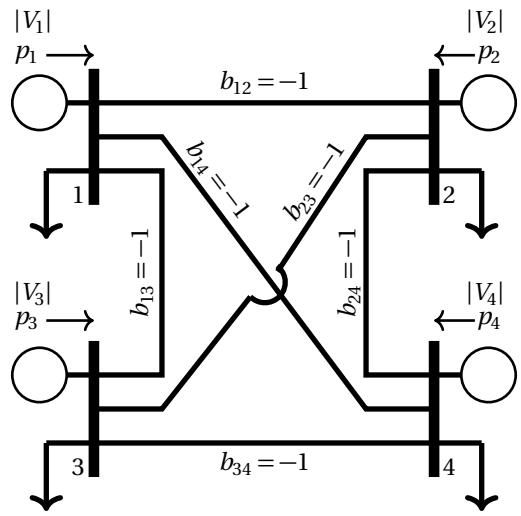


Figure 4.1 One-line diagram for a four-bus electric power system.

- *The case with $\sigma = \iota^{-1}(3)$ produces no roots. Theorem 4.1.17 gives case $j = -1$ as the next case to be considered, so the algorithm stops.*

In this example Theorem 4.1.17 allowed us to skip five of the nine cases that could have possibly been skipped since they gave no equilibria.

In some cases, it is possible to prune more efficiently than Theorem 4.1.17. Consider the following input condition.

IC4: $k_1 \geq k_2 \geq \cdots \geq k_n$

Note that it is not always possible to satisfy both **IC3** and **IC4**, but if an ω and k do satisfy both, then a more efficient algorithm can be used.

Proposition 4.1.23 *Suppose that **IC1–4** are satisfied and u and v are such that $u < v$, $\sigma_u = +$, and $\sigma_v = -$ for some $\sigma \in \{-,+\}$. Let σ' be the same as σ except that $\sigma'_u = -$ and $\sigma'_v = +$. If f_σ has no positive roots, then $f_{\sigma'}$ also has no positive roots.*

Proof. From **IC3** and **IC4** and $u < v$, we have

$$k_u \geq k_v \text{ and } \left(\frac{\omega_u}{k_u}\right)^2 \leq \left(\frac{\omega_v}{k_v}\right)^2.$$

For $R \geq \left(\frac{\omega_n}{k_n}\right)^2$,

$$k_u\sqrt{R - \left(\frac{\omega_u}{k_u}\right)^2} \geq k_v\sqrt{R - \left(\frac{\omega_v}{k_v}\right)^2} \text{ so that } \sqrt{k_u^2 R - \omega_u^2} \geq \sqrt{k_v^2 R - \omega_v^2}.$$

Hence, $f_{\sigma'}(R) \leq f_\sigma(R)$. Therefore, the result follows from Lemma 4.1.13. ■

Writing σ as a binary number, Proposition 4.1.23 allows swapping a "0" and a "1" provided the "0" is on the right of "1." Thus, we can once again skip sequential sign cases once a case with no roots is determined.

Theorem 4.1.24 *Assume that* **IC1–4** *are satisfied. Let $\sigma \in \{-,+\}^n$ be such that f_σ has no positive roots and let p be the index of the penultimate negative in σ or zero if there are less than two negatives in σ. Let $\sigma' \in \{-,+\}^n$ be such that*

- $\sigma'_u = \sigma_u$ *for* $u \leq p$.
- $\sigma'_u = -$ *for* $u > p$.

Then $f_{\hat{\sigma}}$ has no positive roots for any $\hat{\sigma} \in \{-,+\}^n$ where $\iota(\sigma') \leq \iota(\hat{\sigma}) \leq \iota(\sigma)$.

Proof. Let σ, σ', and p be as described in the theorem statement above. Let v be the index of the last negative in σ or zero if there are no negatives in σ. Let $\hat{\sigma} \in \{-,+\}^n$ be such that $\iota(\sigma') \leq \iota(\hat{\sigma}) \leq \iota(\sigma)$. Since the bits of $\iota(\sigma)$ and $\iota(\sigma')$ agree up to the p-th position, only the less significant bits to the right of the p-th position of $\iota(\hat{\sigma})$ can differ from $\iota(\sigma)$. There are two cases to consider.

- $v = 0$ or $\hat{\sigma}_v = -$

  By selection of p and v, we have that $\hat{\sigma}_u \leq \sigma_u$ for $u > p$, $u \neq v$. Thus, Proposition 4.1.14 can be repeatedly used on the appropriate positions just like in Theorem 4.1.17.

- $\hat{\sigma}_v = +$

  Since $\iota(\hat{\sigma}) \leq \iota(\sigma)$, we must have $p < u < v$ such that $\hat{\sigma}_u = -$. Thus we can apply Proposition 4.1.23 to σ_u and σ_v to switch the negative to the left and then use Proposition 4.1.14 to change the remaining differing positions from positive to negative.

■

Theorem 4.1.24 allows us to use the pruning function P_2 to determine the next sign case to be considered. Note that this Theorem 4.1.17 using P_2 is strictly better than Theorem 4.1.17 which used P_1. If a case is determined to produce no solutions, then all of the bits past the penultimate zero can be zeroed out instead of just the bits past the last zero. Thus if **IC1–4** are satisfied, we can replace the pruning in Algorithm 4.1.21 with the stronger Theorem 4.1.24.

Algorithm 4.1.25 (Prune 2)

In: $\omega \in \mathbb{R}^n$ *and* $k \in \mathbb{R}^n_{>0}$ *satisfying* **IC1–IC4**.

Out: $\Theta_{k,\omega}$

1. $\Theta_{k,\omega} \leftarrow \{\}$
2. $j \leftarrow 2^n - 1$

3. *While* $j > 0$ *do*

   (a) $\sigma \leftarrow \iota^{-1}(j)$

   (b) $I \leftarrow \left[\left(\frac{\omega_n}{k_n} \right)^2, \left(\frac{1}{n} \sum_{u \in \sigma_+} k_u \right)^2 \right]$

   (c) *If* I *is empty, then*

      i. $j \leftarrow \mathrm{P}_2(j)$

      ii. *Continue (go back to Step 3)*

   (d) $f_\sigma \leftarrow -R + \frac{1}{n} \sum_{u=1}^{n} \sigma_u \sqrt{k_u^2 R - \omega_u^2}$

   (e) $\mathscr{R} \leftarrow \mathbf{Solve}(f_\sigma, I)$

   (f) *If* $\mathscr{R} = \emptyset$, *then*

      i. $j \leftarrow \mathrm{P}_2(j)$

      ii. *Continue (go back to Step 3)*

   (g) *For* $R \in \mathscr{R}$ *do*

      i. *Compute* $\theta \in (-\pi, \pi]^n$ *according to Proposition 4.1.9*

      ii. *Add* θ *to* $\Theta_{k,\omega}$

   (h) $j \leftarrow j - 1$

Example 4.1.26 *To illustrate, suppose the input parameters satisfy* **IC1–4** *and* f_σ *has no positive roots for* $\sigma = (+1,+1,-1,+1,-1,+1) \underset{\iota}{\mapsto} 53$. *Theorem 4.1.24 shows that* $f_{\sigma'}$ *also has no positive roots for the following sequential sign patterns:*

$$(+1,+1,-1,+1,-1,-1) \underset{\iota}{\mapsto} 52$$
$$(+1,+1,-1,-1,+1,+1) \underset{\iota}{\mapsto} 51$$
$$(+1,+1,-1,-1,+1,-1) \underset{\iota}{\mapsto} 50$$
$$(+1,+1,-1,-1,-1,+1) \underset{\iota}{\mapsto} 49$$
$$(+1,+1,-1,-1,-1,-1) \underset{\iota}{\mapsto} 48$$

Furthermore, 48 *can be immediately calculated from* 53 *by zeroing out everything from the next to last zero onward, so that the five listed cases do not need to be considered at all. Note also that Theorem 4.1.17 would skip only case 52.*

4.1.1 Performance Comparison

A program that combines Algorithm 4.1.21 and Algorithm 4.1.25 was written in C++ using the C-XSC library which is described in [30]. The C-XSC library uses an interval Newton method (see

Chap. 6 of [24]) as its univariate solver for **Solve**. This program was presented in [15] and was used to compare computation times with other common methods. That implementation is available at http://dx.doi.org/10.7274/R09W0CDP as “main.cpp". A similar C++ program is provided in Appendix A for convenience. However, note that this version differs slightly from the implementation used in [15] which was used to generate the timings in this section so as to match the format of this chapter. This section contains those performance comparisons against the following methods:

- solve Eq. 4.0.2 (converted to a polynomial system using trigonometric identities) using Gröbner basis techniques in Macaulay2 [23];
- solve Eq. 4.0.2 (converted to a polynomial system using trigonometric identities) using homotopy continuation in Bertini [4] as in [38];
- compute equilibria of Eq. 4.0.1 using elliptical continuation from [33].

We end with an example having $n = 60$ that is easily solvable using Algorithm 4.1.25.

Comparison with computational algebraic geometry:

We use the following setup from [38] to compare solving Eq. 4.0.2 using Macaulay2 and Bertini with serial computations to Algorithm 4.1.25. For each $n = 3, \ldots, 12$, the natural frequencies are equidistant, namely $\omega_u = -1 + (2u-1)/n$ for $u = 1, \ldots, n$, with uniform coupling $k = (\sqrt{1.5}, \ldots, \sqrt{1.5})$. To simplify the algebraic geometry computations using Macaulay2 and Bertini, we compute the equilibria as in [38] by setting $\theta_n = 0$ ($s_n = 0$ and $c_n = 1$) with the results summarized in Table 4.1.

With Macaulay2, we simply computed the total number of complex solutions, i.e., the degree of the ideal generated by the polynomials when $s_n = 0$ and $c_n = 1$. Thus, one would need to perform additional computations to compute the number of real solutions. The symbol ‡ means that the computation did not complete within 48 hours.

With Bertini, we performed two different computations. The first was to directly solve the polynomial reformulation of Eq. 4.0.2 using regeneration [25] and the second utilized a parameter homotopy [42]. Both of these computations provide all real and non-real solutions to the polynomial reformulation of Eq. 4.0.2.

Although Bertini is parallelized and Algorithm 4.1.25 is parallelizable, we again note that the data in Table 4.1 is based on using serial processing. Nonetheless, this shows the advantage of using Algorithm 4.1.25 to compute all equilibria without needing to compute the non-real solutions of Eq. 4.0.2.

Comparison with elliptical continuation:

We next compare Algorithm 4.1.25 with the elliptical continuation method proposed in [33]. While having the advantage of being applicable to a more general setting of power flow equations, the elliptical continuation method in [33] comes with both theoretical and computational drawbacks

Table 4.1 Time comparison of various solving methods on the rank one Kuramoto model.

n	3	4	5	6	7	8	9	10	11	12
# real	2	2	4	4	4	4	4	4	8	8
# complex	6	12	28	56	118	238	486	976	1972	3958
Macaulay2 degree	< 0.1s	< 0.1s	0.1s	1.1s	7.0s	72.6s	716.5s	10783.7s	149578.0s	‡
Bertini regeneration	0.3s	1.2s	3.4s	13.4s	45.1s	116.6s	210.1s	486.2s	1493.1s	3443.5s
Bertini parameter	< 0.1s	< 0.1s	0.2s	0.4s	1.1s	2.2s	6.9s	15.0s	36.9s	116.8s
Algorithm 4.1.25	< 0.1s	< 0.1s	< 0.1s	< 0.1s	< 0.1s	< 0.1s	< 0.1s	< 0.1s	< 0.1s	< 0.1s

relative to Algorithm 4.1.25 when considered in the context of the Kuramoto model. In contrast to Algorithm 4.1.25, there currently is no theoretical guarantee that the elliptical continuation method in [33] will compute all equilibria. Moreover, the computational speed of Algorithm 4.1.25 can be several orders of magnitude faster than the elliptical continuation method in [33]. Consider, for instance, a test case with $n = 18$, $k = (1, \ldots, 1)$, and

$$\begin{aligned}\omega \quad = \quad & (0.1000, -0.1000, -0.1415, -0.1429, 0.1500, 0.2000, -0.4142, 0.7000, -0.8500, \\ & 1.4142, 2.3000, 3.1415, -3.1904, -3.5000, 4.3333, -5.0000, -6.0000, 7.0000).\end{aligned}$$

When interpreted as a power flow problem, this test case represents a power system composed of 18 buses with fixed, unity voltage magnitudes and specified active power injections given by ω in normalized "per unit" values. The buses are completely connected by lines with unity reactance and zero resistance. While this is a very special example of a power system network, the corresponding test case enables comparison between Algorithm 4.1.25 and the elliptical continuation method in [33] in the context of the Kuramoto model.

A serial implementation of the elliptical continuation method in [33] in Matlab yielded 8538 equilibria satisfying Eq. 4.0.1 in 1.935×10^5 seconds (53.77 hours). For a fair comparison, we used a serial implementation of Algorithm 4.1.25 in Matlab which computed 8538 equilibria in 13.9 seconds. Hence, the implementation of Algorithm 4.1.25 in Matlab is roughly four orders of magnitude faster than the Matlab implementation of [33] for this example. We note that the C++ implementation of Algorithm 4.1.25 took 6.6 seconds.

An example with $n = 60$:

We conclude with an example solved by Algorithm 4.1.25 for $n = 60$ having $k = (60, \ldots, 60)$ and

$$\begin{aligned}\omega = (\quad & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 0, & 20, \\ & -20, & 40, & -60, & 60, & 60, & 80, & -80, & -100, & -100, & 120, \\ & -160, & -160, & -200, & 240, & -280, & -300, & 300, & -360, & 360, & -380, \\ & 420, & 420, & -420, & -460, & 460, & 500, & 520, & 540, & -560, & -600, \\ & -620, & 620, & -640, & 660, & 660, & 660, & 680, & -720, & 780, & -800, \\ & 820, & -820, & -840, & -840, & -880, & 920, & -980, & -980, & -1080, & 3500 \quad).\end{aligned}$$

This example has 2 equilibria satisfying Eq. 4.0.1 with the total computation time using the C++ implementation of Algorithm 4.1.25 taking under a second. For comparison, the elliptical continuation method as described in the previous example took 5609 seconds (93.5 minutes). This example is simply too large for current methods that compute all complex roots. Generally, problems with $\left(\frac{\omega_n}{k_n}\right)^2$ near $\left(\frac{1}{n}\sum_{u=1}^{n} k_u\right)^2$ will be solved quickly by Algorithm 4.1.25 as a consequence of Proposition 4.1.10 and Theorem 4.1.24.

4.1.2 Performance Analysis

This section considers the effectiveness of Algorithms 4.1.21 (Prune 1) and 4.1.25 (Prune 2). We will show that restricting the pruning to sequential sign cases is very effective at using the results of Propositions 4.1.14 and 4.1.23. One way to visualize these comparisons is to create a lattice-like structure where the sign cases are the nodes and the edges show which cases can be ruled out should a case not have any solutions. Examples 4.1.27 and 4.1.28 show the comparisons between the Algorithms' and the Propositions' results.

Example 4.1.27 *Consider the case where $n = 4$. Using the results of Proposition 4.1.14 with the binary enumeration scheme explained in Section 4.1 allows us to create a graph that shows what cases can be skipped once a case with no solutions has been found. (Note that the all negative case which corresponds with zero is not shown as it can never have a solution.)*

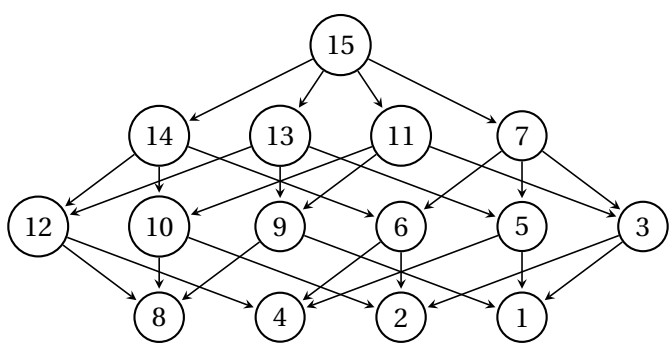


Figure 4.2 Representation of the pruning using Proposition 4.1.14 when $n = 4$.

For example, in Figure 4.2, if the sign case corresponding with 14 *does not have any solutions, then the sign cases* 12, 10, *and* 6 *can be skipped, and the cases that follow from them, namely* 8, 4, *and* 2, *can also be skipped. Restricting the pruning to skip only sequential cases as described by Theorem 4.1.17 gives a subgraph which is shown for $n = 4$ in Figure 4.3.*

If the sign cases corresponding to 15, 14, 13, 12 *and* 11 *are the only ones having roots, then an algorithm that uses the full pruning result of Proposition 4.1.14 and Algorithm 4.1.21 (Prune 1) which*

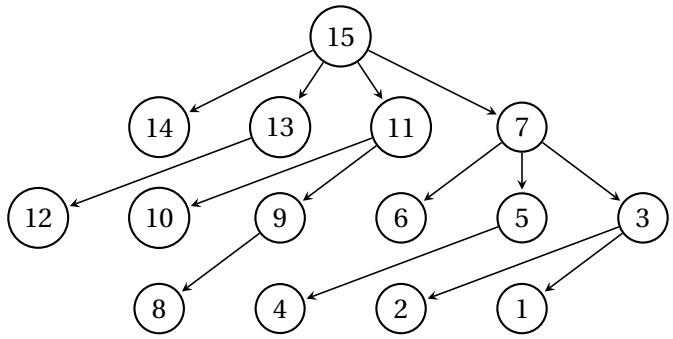


Figure 4.3 The pruning when $n = 4$ described by Theorem 4.1.17 which uses a sequential restriction. This is a subgraph of Figure 4.2.

uses Theorem 4.1.17 are shown for comparison in Figure 4.4. In both cases, three unneccessary cases are checked.

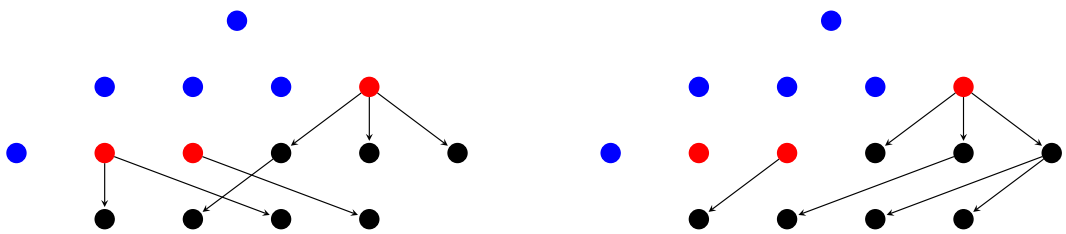


Figure 4.4 The results of an algorithm using the full, nonsequential pruning of Proposition 4.1.14 on the left, and the result of Algorithm 4.1.21 (Prune 1) on the right. The dots correspond to the nodes in Figures 4.2 and 4.3. Blue dots represent cases that were checked and have a root, red dots are cases that were checked and do not have a root, and black dots are cases that are ruled out. Note that only the first edge that rules out a case from consideration is shown.

Example 4.1.28 *Consider the case where $n = 4$ when **IC4** is satisfied. Using the results of Proposition 4.1.23 with the binary enumeration scheme explained in Section 4.1 allows us to create a graph that shows what cases can be skipped once a case with no solutions has been found. The solid edges in Figure 4.5 show these relations. Combining this with Proposition 4.1.14 gives the total possible pruning. The dashed edges in Figure 4.5 represent the pruning from the Proposition 4.1.14 which is not redundant. (Note that the all negative case which corresponds with zero is not shown as it can never have a solution.)*

Restricting the pruning to use sequential cases as described in Theorem 4.1.24 and used in Algorithm 4.1.25 (Prune 2) gives a subgraph which is shown for the $n = 4$ case in Figure 4.6.

For comparison, if the sign cases corresponding to 15, 14, 13, 12 and 11 are the only ones having roots, then an algorithm that uses the full pruning results of Propositions 4.1.14 and 4.1.23 and Algorithm 4.1.25 (Prune 2) which uses Theorem 4.1.24 are shown in Figure 4.7. In both cases, two unneccessary cases are checked.

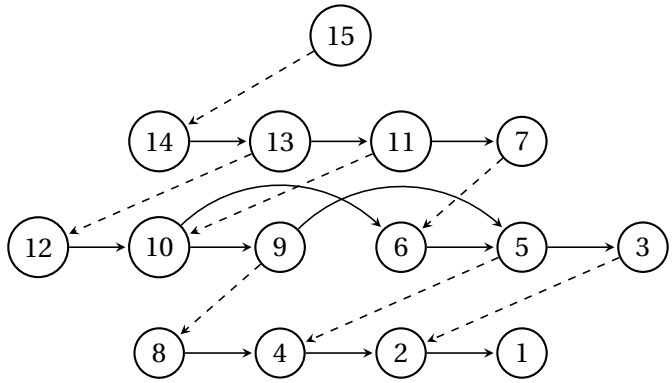


Figure 4.5 Representation of the pruning using Propositions 4.1.14 (dashed arrows) and 4.1.23 (solid arrows). Redundant edges from Proposition 4.1.14, such as from 15 to 13, have been removed for clarity.

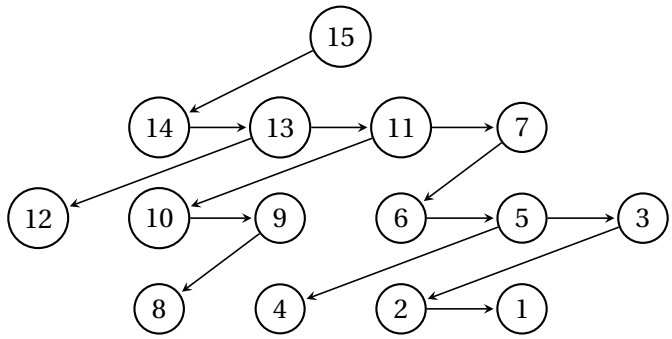


Figure 4.6 The pruning when $n = 4$ described by Theorem 4.1.24 which uses a sequential restriction. This is a subgraph of Figure 4.5.

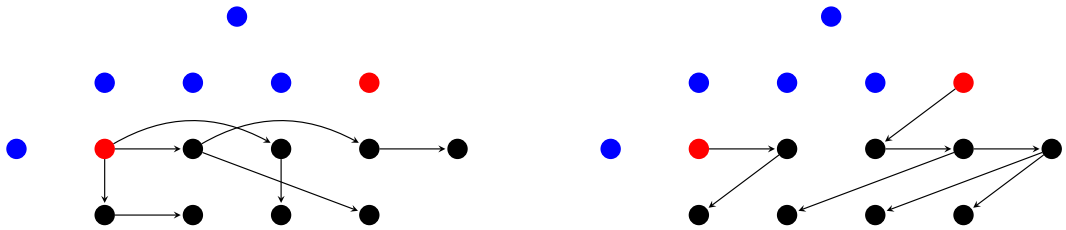


Figure 4.7 The results of an algorithm using the full, nonsequential pruning of Propositions 4.1.14 and 4.1.23 on the left, and the result of Algorithm 4.1.25 (Prune 2) on the right. The dots correspond to the nodes in Figures 4.5 and 4.6. Blue dots represent cases that were checked and have a root, red dots are cases that were checked and do not have a root, and black dots are cases that are ruled out. Note that only the first edge that rules out a case from consideration is shown.

Since if there are no solutions, both Algorithms check only one case, namely the all positive case, this section will assume that there is at least one solution. It will be shown that if S is the set of sign cases containing equilibria of Eq. 4.0.1, then the number of sign cases checked by Algorithm 4.1.21 (Prune 1) is at most $(n+1)|S|$ and the number of sign cases checked by Algorithm 4.1.25 (Prune 2) is at most $2|S|$. In other words Prune 1 checks at most $n|S|$ extra cases and Prune 2 checks at most $|S|$ extra cases. Thus Prune 1 and Prune 2 scale extremely efficiently with the number of equilibria. In many applications the number of equilibria is expected to be relatively low. Hence, these algorithms will be much less time consuming compared to the other approaches discussed in the previous section.

Moreover, we can construct two examples that have the same numbers of extra cases checked when using the full, nonsequential pruning from Propositions 4.1.14 and 4.1.23 respectively. Thus, using the sequential pruning restriction gives the same worse case performance as the nonsequential pruning while also giving a simple computation for how to skip cases as opposed to the exponential time work needed to determine the total pruning.

The first step in proving these results is to abstract the process of skipping sign cases.

Definition 4.1.29 *The **efficiency algorithm** E_ℓ is an algorithm that takes as its input a nonempty subset of $\{1,2,\ldots,2^n-1\}$ for some given integers $n \geq 2$ and $\ell > 0$ and outputs a pair of integers (R,C) according to the following steps:*

In: $S \subseteq \{1,2,\ldots,2^n-1\}$ *s.t.* $S \neq \emptyset$

Out: (R,C)

1. $j \leftarrow 2^n-1,\ R \leftarrow 0,\ C \leftarrow 0$
2. *While* $j > 0$ *do*
   (a) $C \leftarrow C+1$
   (b) *If* $j \in S$ *then*
      i. $R \leftarrow R+1$
      ii. $j \leftarrow j-1$
   (c) *Else*
      i. $j \leftarrow \mathrm{P}_\ell(j)$

*For a given ℓ and n, an input $S \subseteq \{1,2,\ldots,2^n-1\}$, $S \neq \emptyset$ is called **valid** if $R = |S|$ where (R,C) is the output of $\mathrm{E}_\ell(S)$. Let S_ℓ be the set of all valid inputs.*

Lemma 4.1.30 $2^n-1 \in S$ *for any* $S \in S_\ell$.

Proof. Note that the n digit binary representation of 2^n-1 is all ones. Therefore $\mathrm{P}_\ell(2^n-1)=0$ for any $\ell > 0$. Suppose that $2^n-1 \notin S$ for some $S \in S_\ell$. Then $\mathrm{E}_\ell(S)=(0,1)$, but $|S|>0$ by definition of S_ℓ. Since this is a contradiction, we must have $2^n-1 \in S$. ■

The outputs of the efficiency algorithm E_ℓ on a valid input S can be interpreted as

- $R =$ the number of cases checked that are required to be checked
- $C =$ the total number of cases that are checked

Comparing the sizes of R and C quantifies the efficiency of the pruning algorithm P_ℓ since the closer C is to R, the less work was wasted checking unnecessary cases. We restrict the input to valid sets since we will assume that the algorithm we are abstracting is correct, i.e., it will not skip over any sign case that does have a solution.

In order to consider the efficiency, we will build a graph that represents all the possible outputs of E_ℓ as all the possible paths through this graph. Building all possible $(R,\ C)$ pairs produced by $\mathrm{E}_\ell(S_\ell)$ can be done recursively by working backwards. Let E_ℓ^m be the same as E_ℓ except that the algorithm starts "in the middle" by initializing u to m in Step 1 of Definition 4.1.29. For a given $S \in S_\ell$, at any case $m < 2^n - 1$, there are two options:

- $m \in S$, in which case we get $\mathrm{E}_\ell^m(S) = (\alpha+1,\ \beta+1)$ where $(\alpha,\ \beta)$ is the result of $\mathrm{E}_\ell^{m-1}(S)$.
- $m \notin S$, in which case we get $\mathrm{E}_\ell^m(S) = (\alpha,\ \beta+1)$ where $(\alpha,\ \beta)$ is the result of $\mathrm{E}_\ell^{\mathrm{P}_\ell(m)}(S)$.

For $m = 2^n - 1$, we have only the first option by Lemma 4.1.30.

To determine the efficiency, we want to show that $\max\limits_{(R,C)\in \mathrm{E}_\ell(S_\ell)} \dfrac{C}{R} = f(n)$. To do so, we can rewrite any (R, C) pair as $(\alpha,\ f(n)\cdot\alpha + p)$ where p is the "par number" of the pair. If $p \le 0$ for every pair, then the ratio is at most $f(n)$, and if $p = 0$ for some pair, then the maximum is exactly $f(n)$. Consider the two cases above again.

- $m \in S$, so we have $\mathrm{E}_\ell^m(S) = (\alpha+1,\ f(n)\cdot(\alpha+1) + (p - f(n) + 1))$ where $(\alpha,\ f(n)\cdot\alpha + p)$ is the result of $\mathrm{E}_\ell^{m-1}(S)$. Therefore the new par number is lower by $f(n) - 1$.
- $m \notin S$, so we get that $\mathrm{E}_\ell^m(S) = (\alpha,\ f(n)\cdot\alpha + (p+1))$ where $(\alpha,\ f(n)\cdot\alpha + p)$ is the result of $\mathrm{E}_\ell^{\mathrm{P}_\ell(m)}(S)$. Therefore the new par number is one higher.

To represent this graphically, we can connect the node $m-1$ to m with an edge having weight $-f(n)+1$ for $m \le 2^n - 1$, and we can connect the node $\mathrm{P}_\ell(m)$ to m with an edge having weight $+1$ for $m < 2^n - 1$. Showing that $\max\limits_{(R,C)\in \mathrm{E}_\ell(S_\ell)} \dfrac{C}{R} = f(n)$ can now be done by showing that the maximum weight path through this graph is zero.

Before beginning to examine the efficiency of P_1 and P_2, several lemmas regarding the binary representation of sign cases are required.

Lemma 4.1.31 *The $k \ge 1$ digit binary representation of $2^{k-1} \le m < 2^k$ has a leading one.*

Proof. 2^{k-1} has the binary representation of a one followed by $k-1$ zeros. $2^k - 1$ has the binary representation of k ones. Thus every number in between must start with a one and have some combination of $k-1$ ones and zeros. ■

Lemma 4.1.32 *The $k \ge 2$ digit binary representation of $2^{k-1} \le m \le 2^k - 2^{k-2} - 1$ starts with "10."*

Proof. 2^{k-1} has the binary representation of a one followed by $k-1$ zeros. $2^k-2^{k-2}-1$ has the binary representation of a "10" followed by $k-2$ ones. Thus every number in between starts with "10" and has some combination of $k-2$ ones and zeros. ■

Lemma 4.1.33 *The $k \geq 3$ digit binary representation of $2^k-2^{k-2}-2^{k-3} \leq m \leq 2^k-2^{k-2}-1$ starts with "101."*

Proof. $2^k-2^{k-2}-2^{k-3}$ has a binary representation that starts with "101" followed by $k-3$ zeros. $2^k-2^{k-2}-1$ has a binary representation that starts with "101" followed by $k-3$ ones. Thus every number in between starts with "101" and has some combination of $k-3$ ones and zeros. ■

Lemma 4.1.34 *The $k \geq 3$ digit binary representation of $2^{k-1} \leq m \leq 2^k-2^{k-2}-2^{k-3}-1$ starts with "100."*

Proof. 2^{k-1} has a binary representation of a one followed by $k-1$ zeros. $2^k-2^{k-2}-2^{k-3}-1$ has a binary representation that starts with "100" and is followed by $k-3$ ones. Thus every number in between starts with "100" and has some combination of $k-3$ ones and zeros. ■

Example 4.1.35 *Let $k=4$.*

- *Lemma 4.1.31: Every $8 \leq m < 16$ starts with a leading one*

$$8 = 1000_2,\ 9 = 1001_2,\ 10 = 1010_2, \ldots, 15 = 1111_2$$

- *Lemma 4.1.32: Every $8 \leq m \leq 11$ starts with "10."*

$$8 = 1000_2,\ 9 = 1001_2,\ 10 = 1010_2,\ 11 = 1011_2$$

- *Lemma 4.1.33: Every $10 \leq m \leq 11$ starts with "101."*

$$10 = 1010_2,\ 11 = 1011_2$$

- *Lemma 4.1.34: Every $8 \leq m \leq 9$ starts with "100."*

$$8 = 1000_2,\ 9 = 1001_2$$

4.1.2.1 Prune 1

We will show that $\max\limits_{(R,C)\in \mathrm{E}_1(S_1)} \dfrac{C}{R} = n+1$.

Example 4.1.36 *Let $n=3$. Then the graph representing E_1 is shown in Figure 4.8. For $S=\{6,7\} \in S_1$, $\mathrm{E}_1(S)$ would start with $j=7$, go to 6, go to 5, skip to 3, then skip to 0 which ends the algorithm,*

returning $(2,4) = (2, 4\cdot 2 - 4)$*, so this particular example finishes four under par for the max ratio* C/R *of* 4*. We can convert the steps of* E_1 *on* S *into a path in the graph above by reversing the order of the* j *values. Traveling from* 0 *to* 3 *on the red edge then to* 5 *on the red edge, then to* 6 *on the blue edge, then to* 7 *on the blue edge gives a total weight of* $1+1+-3+-3=-4$*, the par number for* S*.*

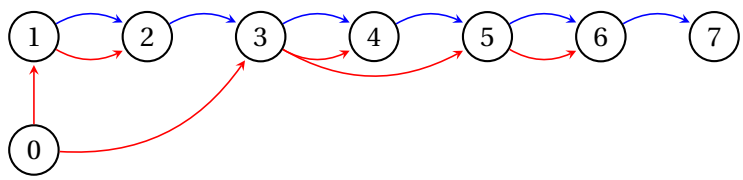


Figure 4.8 The graph representing E_1 when $n=3$. Blue lines have weight -3 and red lines have weight $+1$.

It is useful to break the full graph for E_1 into several parts. Consider the weighted directed graph G_k for $k \geq 2$ with nodes 0, $2^{k-1}-1$, 2^{k-1}, $\ldots$, 2^k-1 with edges based on the following rules.

- An edge with weight $-n$ goes from m to $m+1$ for $2^{k-1}-1 \leq m < 2^k-1$.
- An edge with weight $+1$ goes from $\mathrm{P}_1(m)$ to m for $2^{k-1}-1 \leq m < 2^k-1$.

Note that 0 will have no incoming edges, $2^{k-1}-1$ will have one incoming edge of weight $+1$, 2^{k-1},$\ldots$,2^k-2 will each have two incoming edges, one of weight $-n$ and one of weight $+1$, and 2^k-1 will have one incoming edge of weight $-n$. There are no other edges. Also note that edges go from a smaller number to a larger. Joining G_2, G_3, $\ldots$, G_n together and examining the maximum weight path gives the worst case performance for E_1.

Definition 4.1.37 *Let*

$$I_t^k = 2^k - 1 \qquad II_t^k = 2^k - 2^{k-2} - 2$$
$$I_b^k = 2^k - 2^{k-2} - 1 \qquad II_b^k = 2^{k-1} - 1$$

Then G_k *for* $k \geq 3$ *can be split into two "boxes" as follows:*

- B_1*: The induced subgraph of* G_k *by taking* 0, I_b^k, $\ldots$, I_t^k.
- B_2*: The induced subgraph of* G_k *by taking* 0, II_b^k, $\ldots$, II_t^k.

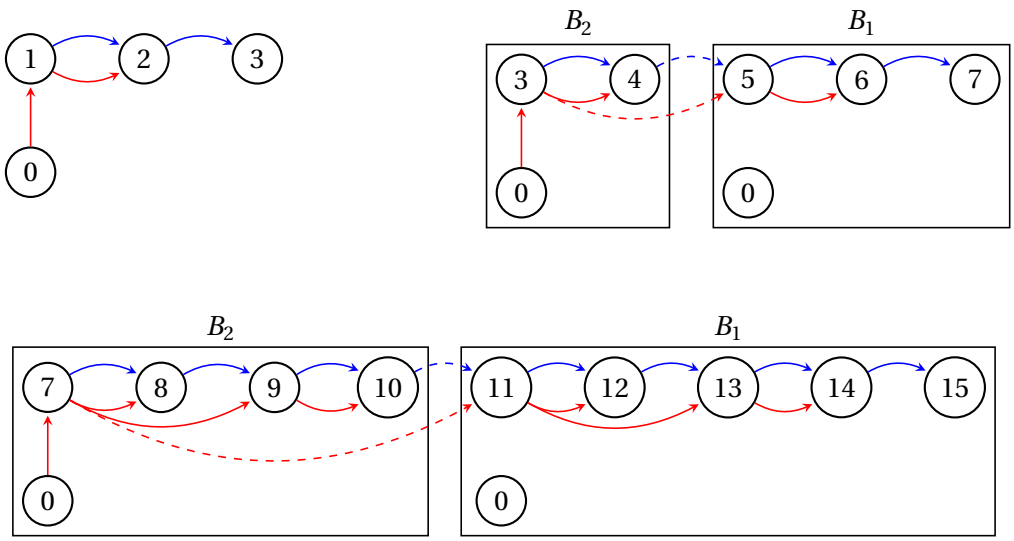


Figure 4.9 G_2, G_3, and G_4 with the boxes described by Definition 4.1.37. Blue edges have weight $-n$ and red edges have weight $+1$. Dashed edges go between boxes. The graph for E_1 when $n = 4$ is $G_2 \cup G_3 \cup G_4$.

Recursive Structure

G_{k+1} can be constructed from G_k using the following propositions.

Proposition 4.1.38 *B_1 of G_{k+1} for $k \geq 2$ is the same as G_k with 2^k added to every nonzero node except that I_b^{k+1} has a blue edge from II_t^{k+1} and a red edge from II_b^{k+1} instead of 0.*

Proof.

- Nodes:

  It is sufficient to show that the top and bottom nodes of G_k with 2^k added match the top and bottom nodes of B_1 of G_{k+1} since all the nodes in between are sequential.

$$\begin{aligned} II_b^k + 2^k &= (2^{k-1} - 1) + 2^k \\ &= 2^{k+1} - (2^{k+1} - 2^k - 2^{k-1}) - 1 \\ &= 2^{k+1} - 2^{k-1} - 1 \\ &= I_b^{k+1} \end{aligned} \qquad \begin{aligned} I_t^k + 2^k &= (2^k - 1) + 2^k \\ &= 2^{k+1} - 1 \\ &= I_t^{k+1} \end{aligned}$$

- Blue Edges:

  $m + 2^{k-1}$ and $(m+1) + 2^{k-1}$ are still adjacent. Also I_b^{k+1} and II_t^{k+1} are adjacent.

- Red Edges:

  Note that for $2^{k-1} \leq m < I_t^k$ the k digit binary representation of m starts with a one by Lemma 4.1.31 and has at least one zero since I_t^k consists of k ones. Adding 2^k to m left-appends a one

to the binary representation. Therefore $\mathrm{P}_1(m+2^k)=\mathrm{P}_1(m)+2^k$ since the left-most one will be unaffected by P_1. Furthermore, I_b^{k+1} has the binary representation of "10" followed by $k-1$ ones. Therefore $\mathrm{P}_1(I_b^{k+1})=I\,I_b^{k+1}$.

■

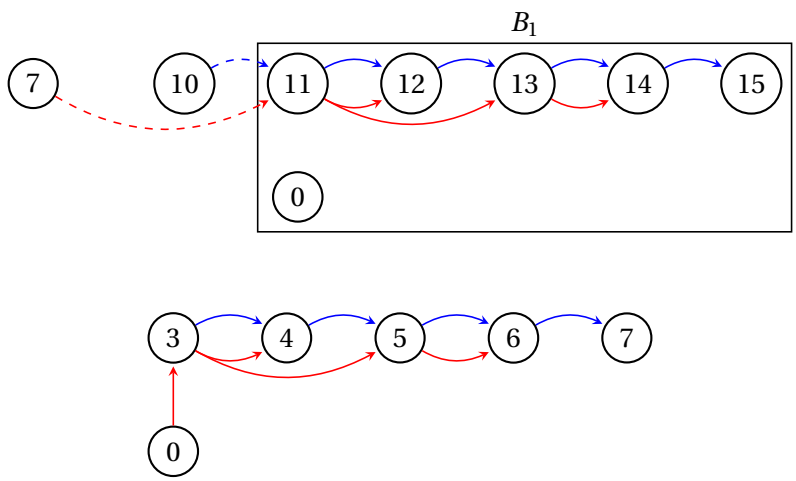


Figure 4.10 B_1 of G_4 and G_3 which have the same structure (see Proposition 4.1.38).

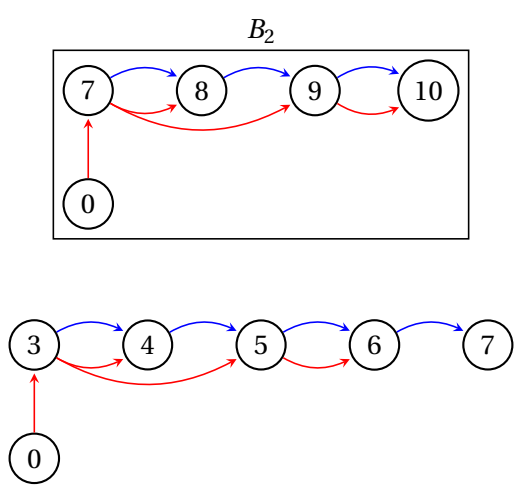


Figure 4.11 B_2 of G_4 and G_3 which have the same structure (see Proposition 4.1.39).

Proposition 4.1.39 *B_2 of G_{k+1} for $k \geq 2$ is the same as $G_k \backslash \left(I_t^k\right)$ with 2^{k-1} added to every nonzero node.*

Proof.

- Nodes:

  It is sufficient to show that the top and bottom nodes of $G_k \backslash \textcircled{I_t^k}$ with 2^{k-1} added match the top and bottom nodes of Box II of G_{k+1} since all the nodes in between are sequential.

$$\begin{aligned} II_b^k + 2^{k-1} &= (2^{k-1}-1)+2^{k-1} \\ &= 2^k - 1 \\ &= II_b^{k+1} \end{aligned} \qquad \begin{aligned} (I_t^k - 1) + 2^{k-1} &= (2^k - 2) + 2^{k-1} \\ &= 2^{k+1} - (2^{k+1} - 2^k - 2^{k-1}) - 2 \\ &= 2^{k+1} - 2^{k-1} - 2 \\ &= II_t^{k+1} \end{aligned}$$

- Blue Edges:

  $\textcircled{m+2^{k-1}}$ and $\textcircled{(m+1)+2^{k-1}}$ are still adjacent.

- Red Edges:

  The k digit binary representation of $II_b^k < m < I_t^k$ has a leading one by Lemma 4.1.31 and at least one zero since I_t^k is k ones. Consider two cases.

  - $\mathrm{P}_1(m)$ has a leading one in its k digit binary representation:
    Then $\mathrm{P}_1(m+2^{k-1}) = \mathrm{P}_1(m) + 2^{k-1}$.
  - $\mathrm{P}_1(m)$ has a leading zero in its k digit binary representation:
    Then m must have the form "10…01…1" and $\mathrm{P}_1(m) = II_b^k$. Therefore, $m+2^{k-1}$ has the form "100…01…1" and $\mathrm{P}_1(m+2^{k-1}) = II_b^{k+1} = II_b^k + 2^{k-1}$.

■

We can make some additional observations about the structure of G_k.

Proposition 4.1.40 *The only edge from* $\textcircled{0}$ *goes to* $\textcircled{II_b^k}$.

Proof. (Proof by induction)

- Base Cases: Observe that this is true for G_2 up to G_4 in Figure 4.9.
- Suppose for induction that the claim holds for G_N where $N \geq 4$ and consider G_{N+1}. By Proposition 4.1.38, B_1 of G_{N+1} is a copy of G_N except that the only edge from $\textcircled{0}$ has been changed. By Proposition 4.1.39, B_2 of G_{N+1} is a copy of $G_N \backslash \textcircled{I_t^k}$, so it will have only one edge from $\textcircled{0}$ which goes to $\textcircled{II_b^{N+1}}$.

■

Proposition 4.1.41 *There are only two edges from B_2 to B_1 of G_k, namely a blue edge from* $\textcircled{II_t^k}$ *to* $\textcircled{I_b^k}$ *and a red edge from* $\textcircled{II_b^k}$ *to* $\textcircled{I_b^k}$.

Proof. Immediate from Proposition 4.1.38. ■

Maximum Weight Path

Now that the structure of the graph has been examined we will use the recursive structure to examine the maximum weight path.

Theorem 4.1.42 *The maximum weight path from* 0 *to* I_t^k *in* G_k *for* $k \geq 2$ *is zero when* $n = k$.

Proof. (Proof by induction)

- Base Cases: The maximum weight path in G_2 up tp G_4 is zero as can be seen Figure 4.9 with $n = k$.
- Suppose for induction that G_N has a maximum weight path of zero and consider G_{N+1}. By Proposition 4.1.40, all paths from 0 to I_t^{N+1} must use the red edge from 0 to II_b^{N+1}. There are two cases to consider by Proposition 4.1.41.
  - A path from II_b^{N+1} to II_t^{N+1} inside B_2 and then the blue edge from II_t^{N+1} to I_b^{N+1} is taken.

    Note that this path is a copy of a path in G_N except that blue edges have a weight that is one less, so the maximum weight is at most zero. Furthermore, a path from I_b^{N+1} to I_t^{N+1} has a maximum weight of at most zero since it is a copy of a path in G_N by Proposition 4.1.38 except with a "cheaper" replacement for the edge from 0 to II_b^N.
  - The red edge from II_b^{N+1} to I_b^{N+1} is taken.

    Any path from I_b^{N+1} to I_t^{N+1} is a copy of a path from II_b^N to I_t^N by Proposition 4.1.38 except that blue edges have a weight that is one less. The blue edge from $I_t^{N+1} - 1$ to I_t^{N+1} must be taken which cancels out the extra red path. Therefore the maximum weight path is zero.

■

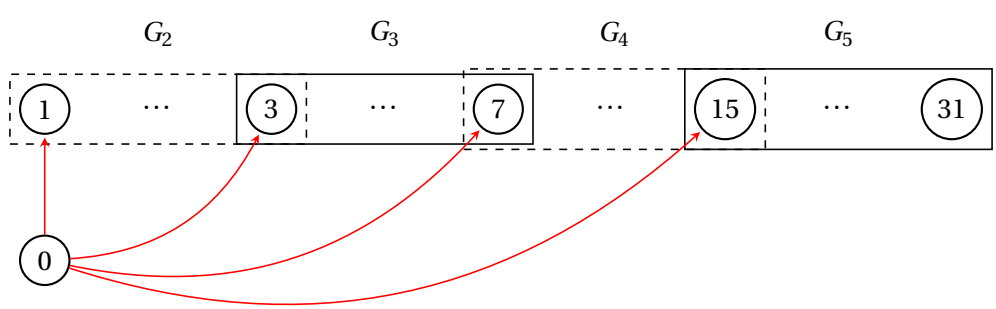


Figure 4.12 $G_2 \cup G_3 \cup G_4 \cup G_5$.

Corollary 4.1.43 *The maximum weight path from* 0 *to* 2^n-1 *in* $G_2 \cup G_3 \cup \cdots \cup G_n$ *is zero.*

Proof. Note that if $n > k$, then the maximum weight path from 0 to I_t^k in G_k is less than zero since all the blue edges are more negative, the red edges are unchanged, and at least one blue edge from $I_t^k - 1$ to I_t^k must be taken. Therefore, taking the edge from 0 to II_b^k and then a path to I_t^k is "cheaper" than the edge from 0 to II_b^{k+1}. Hence the maximum weight path must take the edge from 0 to II_b^n and thus lies entirely within G_n. ■

By construction of the graph for E_1, Corollary 4.1.43 gives us $\max_{(R,C)\in \mathrm{E}_1(S_1)} \frac{C}{R} = n+1$. Thus for any $S \in S_1$, the number of cases checked C is at worst $(n+1)R$. Furthermore, consider the case where $S = \{2^n - 1\}$. Then using the full, nonsequential pruning from Proposition 4.1.14, we would need to check all of the n the cases with one negative before we would have checked or ruled out all the sign cases. Thus the worst case is the same whether using the sequential restriction or not.

4.1.2.2 Prune 2

We will show that $\max_{(R,C)\in E_2(S_2)} \frac{C}{R} = 2$.

Example 4.1.44 *Let* $n = 3$. *Then the graph representing* E_2 *is in Figure 4.13. For* $S = \{6,7\} \in S_2$, $\mathrm{E}_2(S)$ *would start with* $j = 7$, *go to* 6, *go to* 5, *then skip to* 0 *ending the algorithm, returning* $(2,3) = (2, 2\cdot 2 - 1)$, *so this particular example finishes one under par for the max ratio* C/R *of* 2. *We can convert the steps of* E_2 *on* S *into a path in the graph above by reversing the order of the* j *values. Traveling from* 0 *to* 5 *on the red edge, then to* 6 *on the blue edge, then to* 7 *on the blue edge gives a total weight of* $1 + -1 + -1 = -1$, *the par number for* S.

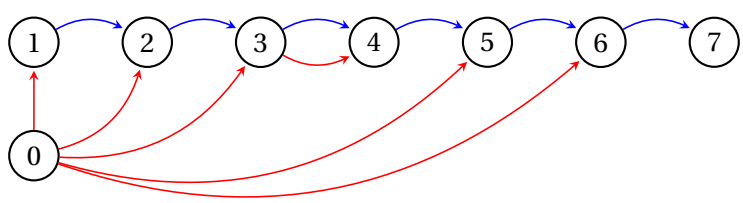


Figure 4.13 The graph representing E_2 when $n = 3$. Blue lines have weight -1 and red lines have weight $+1$.

Again, it is useful to break the graph of E_2 into several parts. Consider the weighted directed graph G_k for $k \geq 2$ with nodes 0, $2^{k-1}-1$, 2^{k-1}, $\ldots$, $2^k - 1$ with edges based on the following rules.

- An edge with weight -1 goes from m to $m+1$ for $2^{k-1} - 1 \leq m < 2^k - 1$.

- An edge with weight $+1$ goes from $\mathrm{P}_2(m)$ to m for $2^{k-1}-1 \le m < 2^k - 1$.

Note that 0 will have no incoming edges, $2^{k-1}-1$ will have one incoming edge of weight $+1$, 2^{k-1},…,2^k-2 will each have two incoming edges, one of weight -1 and one of weight $+1$, and 2^k-1 will have one incoming edge of weight -1. There are no other edges. Also note that edges go from a smaller number to a larger. Joining G_2, G_3, …, G_n together and examining the maximum weight path gives the worst case performance for E_2.

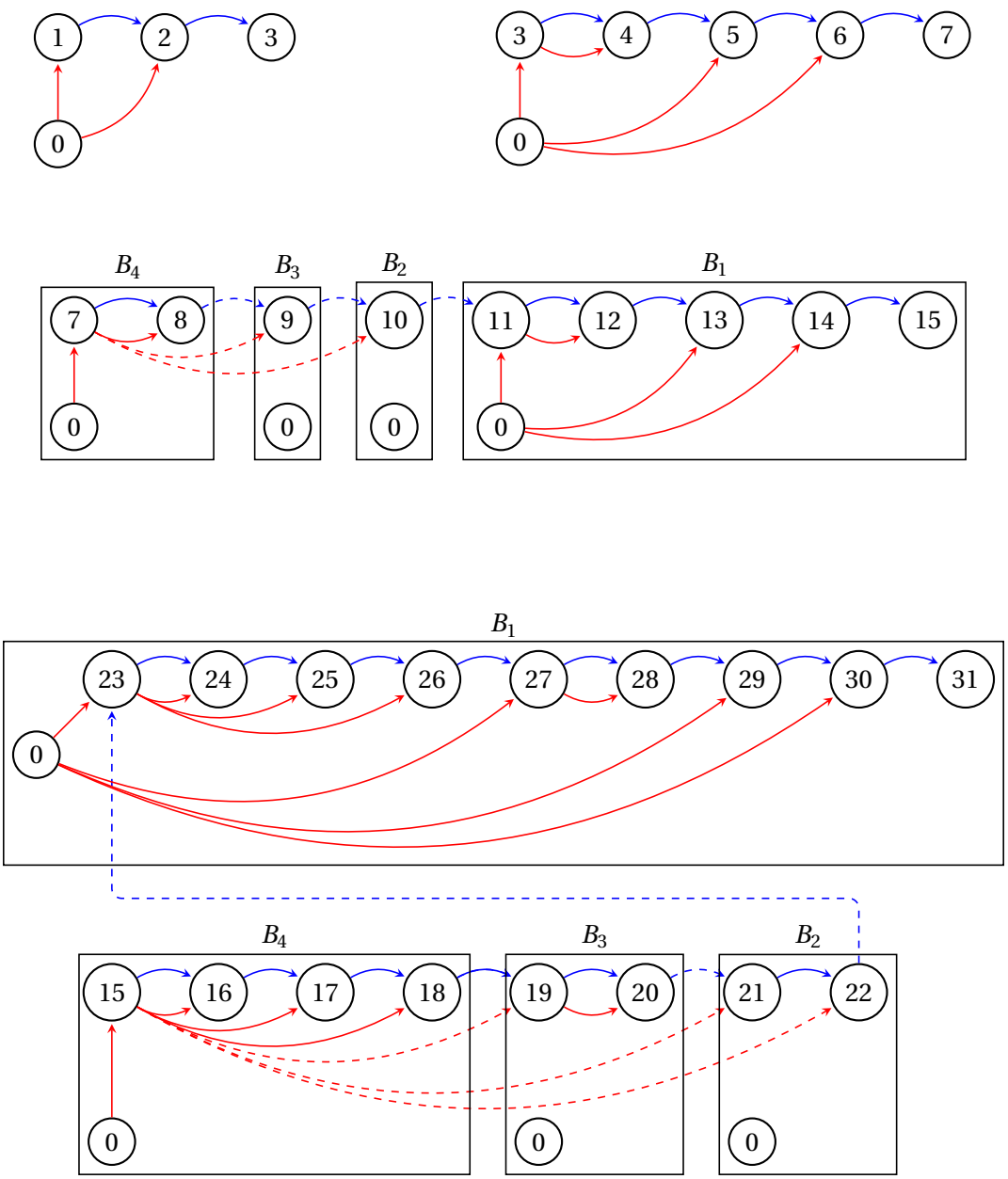


Figure 4.14 G_2, G_3 G_4, and G_5 with the boxes described by Definition 4.1.45. Blue edges have weight -1 and red edges have weight $+1$. Dashed edges go between boxes. The graph for E_2 when $n=5$ is $G_2 \cup G_3 \cup G_4 \cup G_5$.

Definition 4.1.45 *Let*

$$
\begin{aligned}
I_t^k &= 2^k - 1 & III_t^k &= 2^k - 2^{k-2} - 2^{k-4} - 2 \\
I_b^k &= 2^k - 2^{k-2} - 1 & III_b^k &= 2^k - 2^{k-2} - 2^{k-3} - 1 \\
II_t^k &= 2^k - 2^{k-2} - 2 & IV_t^k &= 2^k - 2^{k-2} - 2^{k-3} - 2 \\
II_b^k &= 2^k - 2^{k-2} - 2^{k-4} - 1 & IV_b^k &= 2^{k-1} - 1
\end{aligned}
$$

Then G_k for $k \geq 4$ can be split into four "boxes" as follows:

- B_1: *The induced subgraph of G_k by taking* $0, I_b^k, \ldots, I_t^k$.
- B_2: *The induced subgraph of G_k by taking* $0, II_b^k, \ldots, II_t^k$.
- B_3: *The induced subgraph of G_k by taking* $0, III_b^k, \ldots, III_t^k$.
- B_4: *The induced subgraph of G_k by taking* $0, IV_b^k, \ldots, IV_t^k$.

Recursive Structure

G_{k+1} can be constructed from G_k using the following propositions.

Proposition 4.1.46 *B_1 of G_{k+1} for $k \geq 3$ is the same as G_k with 2^k added to every nonzero node except that I_b^{k+1} has a blue edge from II_t^{k+1}.*

Proof.

- Nodes:

  It is sufficient to show that the top and bottom nodes of G_k with 2^k added match the top and bottom nodes of B_1 of G_{k+1} since all the nodes in between are sequential.

$$
\begin{aligned}
IV_b^k + 2^k &= (2^{k-1} - 1) + 2^k \\
&= 2^{k+1} - (2^{k+1} - 2^k - 2^{k-1}) - 1 \\
&= 2^{k+1} - 2^{k-1} - 1 \\
&= I_b^{k+1}
\end{aligned}
\qquad
\begin{aligned}
I_t^k + 2^k &= (2^k - 1) + 2^k \\
&= 2^{k+1} - 1 \\
&= I_t^{k+1}
\end{aligned}
$$

- Blue Edges:

  $m + 2^k$ and $(m+1) + 2^k$ are still adjacent. Also I_b^{k+1} and II_t^{k+1} are adjacent.

- Red Edges:

  Note that adding 2^k to $0 < m < 2^k$ is the same as left-appending a one to the k digit binary representation of m. Consider two subcases.

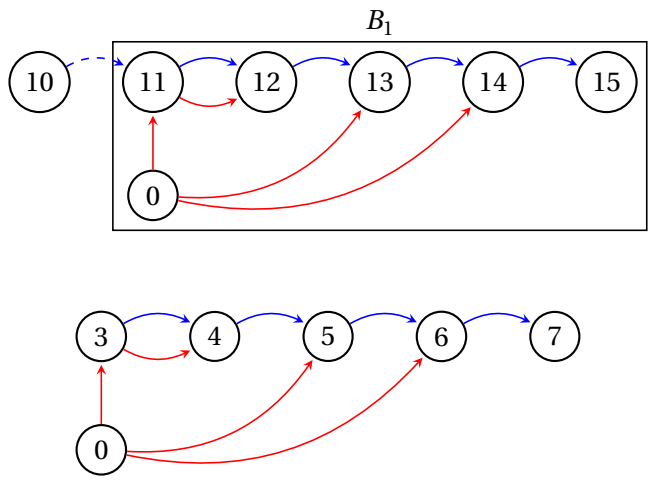


Figure 4.15 B_1 of G_4 and G_3 which have the same structure (see Proposition 4.1.46).

- m has < 2 zeros in its k digit binary representation where $2^{k-1}-1 \le m < 2^k - 1$:
  Then $\mathrm{P}_2(m) = \mathrm{P}_2(m+2^k) = 0$.
- m has ≥ 2 zeros in its k digit binary representation where $2^{k-1}-1 \le m < 2^k - 1$:
  By Lemma 4.1.31, $m+2^k$ has a binary representation that starts with "11." Thus $\mathrm{P}_2(m+2^k) = 2^k + \mathrm{P}_2(m)$ since the left-most one is unaffected by P_2.

■

Proposition 4.1.47 *$B_2 \cup \left(IV_b^{k+1}\right)$ of G_{k+1} for $k \ge 4$ is the same as $B_2 \cup B_3 \cup \left(IV_b^k\right)$ of G_k with $3 \cdot 2^{k-2}$ added to every node $III_b^k \le m \le III_t^k$ and adding 2^{k-1} to IV_b^k.*

Proof.

- Nodes:

  It is sufficient to show that the top and bottom nodes of $B_2 \cup B_3$ of G_k with $3 \cdot 2^{k-2}$ added match the top and bottom nodes of B_2 of G_{k+1}, since all the nodes in between are sequential, and that $IV_b^k + 2^{k-1} = IV_b^{k+1}$.

$$\begin{aligned} III_b^k + 3 \cdot 2^{k-2} &= (2^k - 2^{k-2} - 2^{k-3} - 1) + 3 \cdot 2^{k-2} \\ &= (2^k - 2^{k-2} - 2^{k-3} - 1) + (2^{k-1} + 2^{k-2}) \\ &= 2^{k+1} - (2^{k+1} - 2^k - 2^{k-1}) - 2^{k-3} - 1 \\ &= 2^{k+1} - 2^{k-1} - 2^{k-3} - 1 \\ &= III_b^{k+1} \end{aligned}$$

$$\begin{aligned} III_t^k + 3 \cdot 2^{k-2} &= (2^k - 2^{k-2} - 2) + 3 \cdot 2^{k-2} \\ &= (2^k - 2^{k-2} - 2) + (2^{k-1} + 2^{k-2}) \end{aligned}$$

$$
\begin{aligned}
&= 2^{k+1} - (2^{k+1} - 2^k - 2^{k-1}) - 2 \\
&= 2^{k+1} - 2^{k-1} - 2 \\
&= II_t^{k+1}
\end{aligned}
$$

$$
\begin{aligned}
IV_b^k + 2^{k-1} &= (2^{k-1} - 1) + 2^{k-1} \\
&= 2^k - 1 \\
&= IV_b^{k+1}
\end{aligned}
$$

- Blue Edges:

  $m + 3 \cdot 2^{k-2}$ and $(m+1) + 3 \cdot 2^{k-2}$ are still adjacent.

- Red Edges:

  By Lemma 4.1.33, every node in $B_2 \cup B_3$ of G_k except III_b^k starts with "101" and has at least one more zero. Adding $3 \cdot 2^{k-2} = 2^{k-1} + 2^k$ means that these nodes now have $k+1$ digits and start with "1011." Consider two subcases.

  - m has 2 zeros in its k digit binary representation where $III_b^k < m \leq II_t^k$:
    Then $\mathrm{P}_2(m) = IV_b^k$ and $\mathrm{P}_2(m + 3 \cdot 2^{k-2}) = IV_b^{k+1}$.
  - m has > 2 zeros in its k digit binary representation where $III_b^k < m \leq II_t^k$:
    Then $\mathrm{P}_2(m + 3 \cdot 2^{k-2}) = \mathrm{P}_2(m) + 3 \cdot 2^{k-2}$ since the leading "10" of m is unaffected by P_2.

  By Lemma 4.1.34, $III_b^k + 3 \cdot 2^{k-2} = III_b^k + 2^{k-1} + 2^{k-2}$ starts with "1010" followed by $k-3$ ones. Thus $\mathrm{P}_2(III_b^k) = IV_b^k$ and $\mathrm{P}_2(III_b^k + 3 \cdot 2^{k-2}) = IV_b^{k+1}$.

■

Proposition 4.1.48 *B_3 of G_{k+1} for $k \geq 4$ is the same as B_4 of G_k with $3 \cdot 2^{k-2}$ added to every node $IV_b^k \leq m \leq IV_t^k$ except III_b^{k+1} has a red edge from IV_b^{k+1} instead of 0 and a new blue edge from IV_t^{k+1}.*

Proof.

- Nodes:

  It is sufficient to show that the top and bottom nodes of B_4 of G_k with $3 \cdot 2^{k-2}$ added match the top and bottom nodes of B_3 of G_{k+1} since all the nodes in between are sequential.

$$
\begin{aligned}
IV_b^k + 3 \cdot 2^{k-2} &= (2^{k-1} - 1) + 3 \cdot 2^{k-2} \\
&= (2^{k-1} - 1) + (2^{k-1} + 2^{k-2}) \\
&= 2^k + 2^{k-2} - 1 \\
&= 2^{k+1} - (2^{k+1} - 2^k) + 2^{k-2} - 1
\end{aligned}
$$

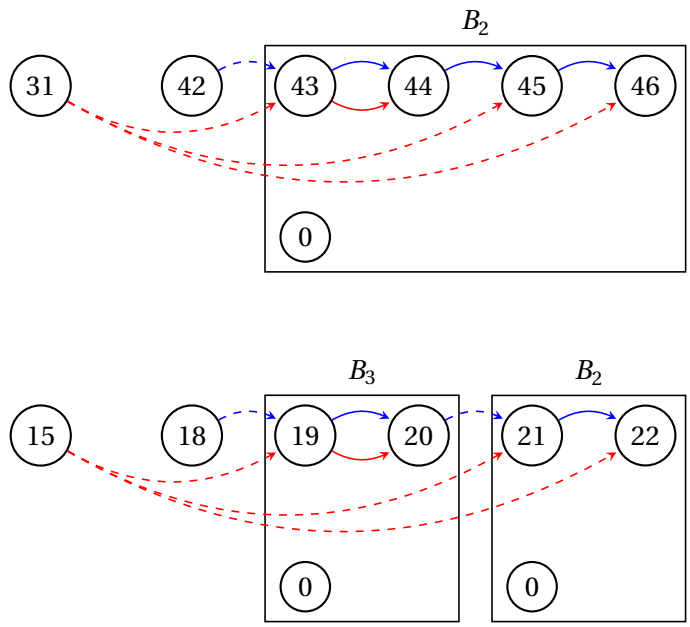


Figure 4.16 B_2 of G_6 and $B_2 \cup B_3$ of G_5 which have the same structure (see Proposition 4.1.47).

$$
\begin{aligned}
&= 2^{k+1} - 2^k + 2^{k-2} - 1 \\
&= 2^{k+1} - (2^{k-1} + 2^{k-1}) + 2^{k-2} - 1 \\
&= 2^{k+1} - 2^{k-1} - (2^{k-1} - 2^{k-2}) - 1 \\
&= 2^{k+1} - 2^{k-1} - 2^{k-2} - 1 \\
&= III_b^{k+1}
\end{aligned}
$$

$$
\begin{aligned}
IV_t^k + 3 \cdot 2^{k-2} &= (2^k - 2^{k-2} - 2^{k-3} - 2) + 3 \cdot 2^{k-2} \\
&= (2^k - 2^{k-2} - 2^{k-3} - 2) + (2^{k-1} + 2^{k-2}) \\
&= 2^k + 2^{k-1} - 2^{k-3} - 2 \\
&= 2^{k+1} - (2^{k+1} - 2^k - 2^{k-1}) - 2^{k-3} - 2 \\
&= 2^{k+1} - 2^{k-1} - 2^{k-3} - 2 \\
&= III_t^{k+1}
\end{aligned}
$$

- Blue Edges:

  $m + 3 \cdot 2^{k-2}$ and $(m+1) + 3 \cdot 2^{k-2}$ are still adjacent. Also, IV_t^{k+1} and III_b^{k+1} are adjacent.

- Red Edges:

  By Lemma 4.1.34, the k digit binary representation for $2^{k-1} \le m \le IV_t^k$ starts with "100." Thus $m + 3 \cdot 2^{k-2}$ has $k+1$ digits and starts with "1010." Consider two cases.

  - The k digit binary representation of $P_2(m)$ for $2^{k-1} \le m \le IV_t^k$ starts with "100":

Then $P_2(m+3\cdot 2^{k-2}) = P_2(m)+3\cdot 2^{k-2}$.

- The k digit binary representation of $P_2(m)$ for $2^{k-1} \le m \le IV_t^k$ is "$011\ldots1$" $= IV_b^k$:
  Then $P_2(m+3\cdot 2^{k-2}) =$ "$10011\ldots1$" $= III_b^{k+1}$.

Also, $P_2(III_b^{k+1}) =$ "$011\ldots1 = IV_b^{k+1}$.

■

Proposition 4.1.49 *B_4 of G_{k+1} for $k \ge 4$ is the same as $B_2 \cup B_3 \cup B_4$ of G_k with 2^{k-1} added to every nonzero node.*

Proof.

- Nodes:

  It is sufficient to show that the top and bottom nodes of $B_2 \cup B_3 \cup B_4$ of G_k with 2^{k-1} added match the top and bottom nodes of B_4 of G_{k+1} since all the nodes in between are sequential.

$$\begin{aligned} IV_b^k + 2^k &= (2^{k-1}-1)+2^{k-1} \\ &= 2^k - 1 \\ &= IV_b^{k+1} \end{aligned}$$

$$\begin{aligned} II_t^k + 2^{k-1} &= (2^k - 2^{k-2} - 2) + 2^{k-1} \\ &= 2^{k+1} - (2^{k+1} - 2^k - 2^{k-1}) - 2^{k-2} - 2 \\ &= 2^{k+2} - 2^{k-1} - 2^{k-2} - 2 \\ &= IV_t^{k+1} \end{aligned}$$

- Blue Edges:

  $m+2^{k-1}$ and $(m+1)+2^{k-1}$ are still adjacent.

- Red Edges:

  By Lemma 4.1.32, every node in $B_2 \cup B_3 \cup B_4$ of G_k except IV_b^k has k digits and starts with "10." Adding 2^{k-1} means they will have $k+1$ digits and start with "100." Consider two cases.

  - The k digit representation of $P_2(m)$ for $IV_b^k < m \le II_t^k$ starts with "10":
    Then $P_2(m+2^{k-1}) = P_2(m)+2^{k-1}$.
  - The k digit representation of $P_2(m)$ for $IV_b^k < m \le II_t^k$ is "$011\ldots1$" $= IV_b^k$:
    Then $P_2(m+2^{k-1}) = IV_b^{k+1}$.

  Also, $P_2(IV_b^{k+1}) = 0$.

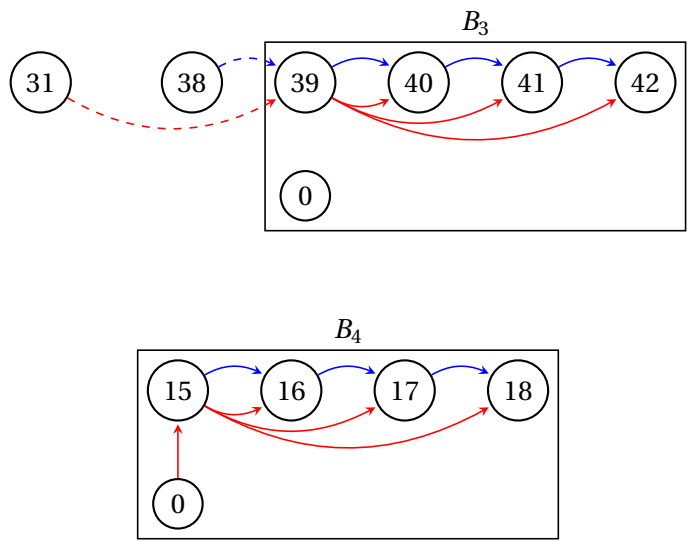


Figure 4.17 B_3 of G_6 and B_4 of G_5 which have the same structure (see Proposition 4.1.48).

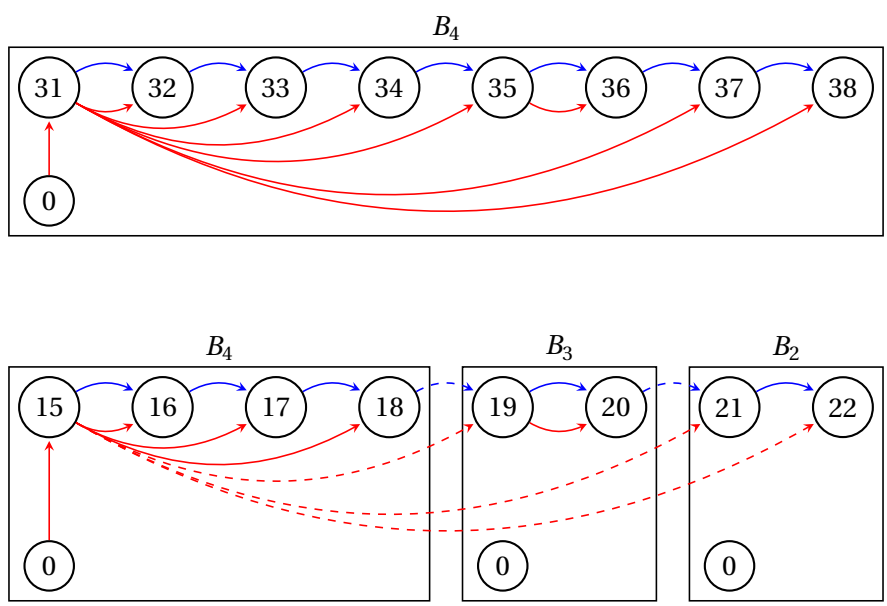


Figure 4.18 B_4 of G_6 and $B_2 \cup B_3 \cup B_4$ of G_5 which have the same structure (see Proposition 4.1.49).

■

We can make some additional observations about the structure of G_k.

Proposition 4.1.50 *$B_2 \cup B_3$ of G_k has no edge from 0 and B_4 of G_k has only one edge from 0 which goes to IV_b^k for $k \geq 4$.*

Proof. (Proof by induction)

- Base case: Observe that this is true for G_4 and G_5 in Figure 4.14.
- Suppose the proposition is true for $k \geq 4$. Consider G_{k+1}.

  By Proposition 4.1.47, B_2 of G_{k+1} is a copy of $B_2 \cup B_3$ of G_k, so will have no edges from 0.

  By Proposition 4.1.48, B_3 of G_{k+1} is a copy of B_4 of G_k except that the (only) edge from 0 has been changed, so it will have no edges from 0.

  By Proposition 4.1.49, B_4 of G_{k+1} is a copy of $B_2 \cup B_3 \cup B_4$ of G_k, so it will have only one edge from 0 which goes to IV_b^{k+1}.

■

Proposition 4.1.51 *B_1 of G_k has only one edge from outside of B_1 which is the edge with weight -1 from II_t^k to I_b^k.*

Proof. Immediate from Proposition 4.1.46. ■

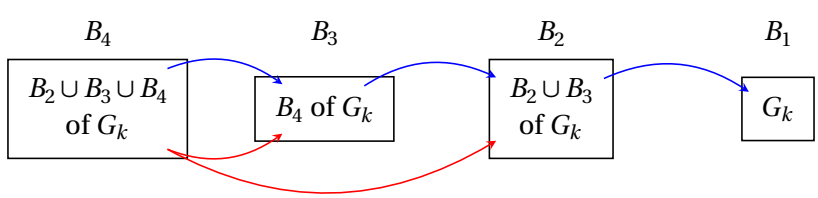


Figure 4.19 Putting Propositions 4.1.46 – 4.1.49 together allows for G_{k+1} to be built from G_k.

Maximum Weight Path

Theorem 4.1.52 *The maximum weight path from 0 to I_t^k in G_k for $k \geq 2$ is zero.*

Proof. (Proof by induction)

- Base cases: Checking G_2 up to G_5 in Figure 4.14 shows that the maximum weight path from 0 to I_t^k is 0 for $k = 2, 3, 4, 5$. Also, the "bottom halves" of G_4 and G_5, i.e., $B_2 \cup B_3 \cup B_4$, have a maximum weight path of zero from IV_b^k to I_b^k.

- Assume for induction that G_k has a maximum weight path of zero and that the maximum weight path from IV_b^k to I_b^k is zero for $k \leq N$ where $N \geq 5$. Consider G_{N+1}. By Propositions 4.1.46 and 4.1.50 there are two choices to travel from 0.
  - A path from 0 to B_1 is taken:
    Then the path is entirely contained inside G_N by Proposition 4.1.46, so the maximum weight is zero by the inductive hypothesis.
  - The path from 0 to IV_b^{N+1} is taken:
    Here again there are two options by Proposition 4.1.51.
    - A path from IV_b^{N+1} to a node in B_4 is taken:
      By Propositions 4.1.47 and 4.1.48, all such paths must go through III_b^{N+1}. By Proposition 4.1.49, this portion of the path is identical to the bottom half of G_N, so by the inductive hypothesis, this portion has a maximum weight of zero. Again, by Propositions 4.1.47 and 4.1.48, the path from III_b^{N+1} to I_b^{N+1} is a copy of the bottom half of G_N except with fewer edges of weight $+1$, thus this portion of the path must have a maximum weight of at most zero. Therefore, 0 to IV_b^{N+1} and then the path from IV_b^{N+1} to I_b^{N+1} has a maximum weight of at most one, which is the same as taking the direct path from 0 to I_b^{N+1}. Since the direct path is completely contained in G_N by Proposition 4.1.46, it results in a maximum weight path of zero. Replacing the direct path with a longer path through the bottom half with the same weight will still result in a maximum weight path of zero.
    - A path from IV_b^{N+1} to a node in B_2 is taken:
      Since $IV_b^{N+1} \cup B_2$ of G_{N+1} is a copy of $IV_b^N \cup B_2 \cup B_3$ of G_N by Proposition 4.1.47, the maximum weight path is zero by the inductive hypothesis.
    - The path from IV_b^{N+1} to III_b^{N+1} in B_3 is taken:
      By Proposition 4.1.48, B_3 of G_{N+1} is a copy of B_4 of G_N. This in turn is a copy of $B_2 \cup B_3 \cup B_4$ of G_{N-1} by Proposition 4.1.49. Thus the maximum weight path from III_b^{N+1} to II_b^{N+1} is zero by the inductive hypothesis. Therefore, taking the edge from IV_b^{N+1} to III_b^{N+1} and then a path from there to II_b^{N+1} has the same weight as taking the edge from IV_b^{N+1} to II_b^{N+1} directly, so the maximum weight path is again zero by the above case.

■

Corollary 4.1.53 *The maximum weight path from* 0 *to* $2^n - 1$ *in* $G_2 \cup G_3 \cup \cdots \cup G_n$ *is zero.*

Proof. Note that taking a red edge from 0 to a node in G_k and then the path from that node to I_t^k has maximum weight zero by the above theorem while the red edge from 0 to I_t^k has a

weight of one. Therefore the maximum weight path must take a red edge from 0 to a node in G_n. Thus the maximum weight path lies entirely in G_n. ■

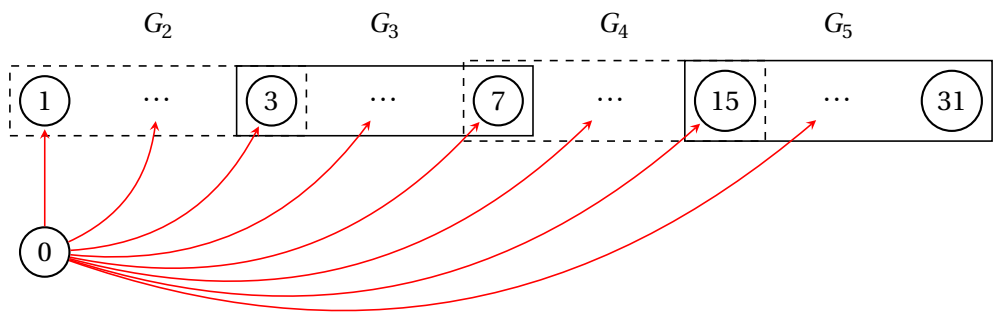


Figure 4.20 $G_2 \cup G_3 \cup G_4 \cup G_5$.

By construction of the graph for E_2, Corollary 4.1.53 gives us $\max_{(R,C)\in \mathrm{E}_2(S_2)} \frac{C}{R} = 2$. Thus for any $S \in S_2$, the number of cases checked C is at worst $2R$. Furthermore, consider the case where $S = \{2^n - 1\}$. Then an algorithm that uses the full, nonsequential pruning of Propositions 4.1.14 and 4.1.23 would check case $2^n - 2$, giving $C = 2R$. Therefore, the worst case efficiency is the same when applying the sequential pruning restriction.

4.2 Stability

In this section we will show that if there is at least one equilibrium, then the rank one Kuramoto model generically has a unique stable equilibria (in the orbitally stable sense; see Section 1.1.2 for a discussion of orbital stability). Furthermore, this stable equilibria can be found from the largest root of f_+. To start, we will examine the Jacobian matrix J. A direct computation of the Jacobian of Eq. 4.0.1 gives

$$J = -\frac{1}{n}\begin{bmatrix} \sum_{v=1}^{n} k_1 k_v \cos(\theta_1 - \theta_v) & & \\ & \ddots & \\ & & \sum_{v=1}^{n} k_n k_v \cos(\theta_n - \theta_v) \end{bmatrix} + \frac{1}{n}\Big[k_u k_v \cos(\theta_u - \theta_v)\Big]_{(u,v)} \tag{4.2.1}$$

Lemma 4.2.1 *The Jacobian matrix has a zero eigenvalue.*

Proof. It is immediate from Eq. 4.2.1 that the sum of all the columns of J is the zero vector. Therefore the all ones vector is an eigenvector with the corresponding eigenvalue of zero. ■

The all ones eigenvector corresponds to the invariance under shift of the Kuramoto model.

Since the model depends only on the differences between angles, we will ignore the shift when determining the stability of an equilibrium. Moreover, note that J is a real symmetric matrix, so all its eigenvalues are real. Thus if J evaluated at an equilibrium has $n-1$ negative eigenvalues, then that equilibrium is stable. Likewise, if J has a positive eigenvalue, then that equilibrium cannot be stable [9].

Remark 4.2.2 *For convenience, we will call $r \in R_{k,\omega,\sigma}$ stable if the corresponding $\theta = F(G_\sigma(r))$ is orbitally stable.*

Here we will introduce some convenient notation.

Notation 4.2.3

- *Let* $\mathbf{+} := (+, \dots, +)$.
- *Let* $s_u := \sqrt{k_u^2 R - \omega_u^2}$.
- *Recall from Section 4.1 that* $f_\sigma(R) := -R + \dfrac{1}{n}\sum_{u=1}^{n} \sigma_u s_u$.

We can convert J using the same variables as the reformulation in Section 4.1. For convenience we will restate some results from the reformulation first.

Lemma 4.2.4 *Let $\phi \in \Phi_{k,\omega,\sigma}$. Then*

$$\mathrm{Im}(\phi_u) = \frac{\omega_u}{k_u\sqrt{R}} \qquad\qquad \mathrm{Re}(\phi_u) = \frac{\sigma_u s_u}{k_u\sqrt{R}}$$

Proof. From Lemma 4.1.2 we have that $\omega_u = k_u r\,\mathrm{Im}(\phi_u)$ where $r \in \mathbb{R}_+$ for any equilibria. We also have $\phi_u = \dfrac{\omega_u i + \sigma_u s_u}{k_u r}$ from Lemma 4.1.4. Lastly, we defined $R := r^2$. ■
(Note as a consequence of this Lemma that $s_u \geq 0$ for all u.)

Proposition 4.2.5 *Let*

$$\phi_u := e^{i\theta_u} \qquad\qquad r := \frac{1}{n}\sum_{u=1}^{n} k_u \phi_u$$

Then $J = D + MM^{\mathrm{T}}$ for any equilibrium of Eq. 4.0.1 where

$$D := -\begin{bmatrix} \sigma_1 s_1 & & \\ & \ddots & \\ & & \sigma_n s_n \end{bmatrix} \qquad\qquad M := \frac{1}{\sqrt{n}}\begin{bmatrix} k_1\,\mathrm{Re}(\phi_1) & k_1\,\mathrm{Im}(\phi_1) \\ \vdots & \vdots \\ k_n\,\mathrm{Re}(\phi_n) & k_n\,\mathrm{Im}(\phi_n) \end{bmatrix}$$

Proof. Applying the change of variables stated in the proposition to Eq. 4.2.1 gives

$$J = -\begin{bmatrix} k_1\,\mathrm{Re}(\phi_1 r^\star) & & \\ & \ddots & \\ & & k_n\,\mathrm{Re}(\phi_n r^\star) \end{bmatrix} + \frac{1}{n}\Big[k_u k_v\,\mathrm{Re}(\phi_u \phi_v^\star)\Big]_{(u,v)}$$

For an equilibrium, we have $r \in \mathbb{R}_+$. Using Lemma 4.2.4 and factoring the second matrix therefore gives

$$
\begin{aligned}
J =& -\begin{bmatrix} k_1 r \operatorname{Re}(\phi_1) & & \\ & \ddots & \\ & & k_n r \operatorname{Re}(\phi_n) \end{bmatrix} \\
&+ \frac{1}{n} \begin{bmatrix} k_1 \operatorname{Re}(\phi_1) & k_1 \operatorname{Im}(\phi_1) \\ \vdots & \vdots \\ k_n \operatorname{Re}(\phi_n) & k_n \operatorname{Im}(\phi_n) \end{bmatrix} \begin{bmatrix} k_1 \operatorname{Re}(\phi_1) & \cdots & k_n \operatorname{Re}(\phi_n) \\ k_1 \operatorname{Im}(\phi_1) & \cdots & k_n \operatorname{Im}(\phi_n) \end{bmatrix} \\
=& -\begin{bmatrix} \sigma_1 s_1 & & \\ & \ddots & \\ & & \sigma_n s_n \end{bmatrix} + \frac{1}{n} \begin{bmatrix} k_1 \operatorname{Re}(\phi_1) & k_1 \operatorname{Im}(\phi_1) \\ \vdots & \vdots \\ k_n \operatorname{Re}(\phi_n) & k_n \operatorname{Im}(\phi_n) \end{bmatrix} \begin{bmatrix} k_1 \operatorname{Re}(\phi_1) & \cdots & k_n \operatorname{Re}(\phi_n) \\ k_1 \operatorname{Im}(\phi_1) & \cdots & k_n \operatorname{Im}(\phi_n) \end{bmatrix} \\
=& D + MM^{\mathrm{T}}
\end{aligned}
$$

■

Next, we can restrict the possible σ values that can produce stable equilibria.

Proposition 4.2.6 *The set of all stable equilibria is a subset of* $F(G_{\mathbf{+}}(R_{k,\omega,\mathbf{+}}))$

Proof. For any $r \in R_{k,\omega,\sigma}$, Lemma 1.2.1 combined with Proposition 4.2.5 gives

$$\lambda_u(J) = \lambda_u(D + MM^{\mathrm{T}}) \geq \lambda_u(D)$$

since $MM^{\mathrm{T}} \succeq 0$. Furthermore, D is diagonal, so its eigenvalues are the diagonal elements $-\sigma_u s_u$. Suppose $\hat{r} \in R_{k,\omega,\sigma}$ is stable and let $\hat{R} = \hat{r}^2$. Then we must have that $-\sigma_u s_u \leq 0$ for every u. Thus for any v such that $\sigma_v = -$, we must have $s_v = 0$. But then $f_\sigma(\hat{R}) = f_{\mathbf{+}}(\hat{R})$, so $\hat{r} \in \mathbb{R}_{k,\omega,\mathbf{+}}$. ■

Recall that the domain of $f_\sigma(R)$ is $\left[\left(\frac{\omega_n}{k_n}\right)^2, \infty\right)$. Thus for $\hat{r} \in R_{k,\omega,\sigma}$, we generically have $\hat{R} = \hat{r}^2 > \left(\frac{\omega_n}{k_n}\right)^2$ which implies $s_u > 0$. By restricting σ to the case $\sigma = \mathbf{+}$, we can determine that $n-2$ of the necessary $n-1$ eigenvalues are negative in this generic case.

Lemma 4.2.7 *For generic* $r \in R_{k,\omega,\mathbf{+}}$, J *has at least* $n-2$ *negative eigenvalues.*

Proof. For any generic $r \in R_{k,\omega,\mathbf{+}}$, Lemma 1.2.1 combined with Proposition 4.2.5 gives

$$\lambda_u(-J) = \lambda_u(-D - MM^{\mathrm{T}}) > \lambda_u(-MM^{\mathrm{T}})$$

since generically $-D \succ 0$ when $\sigma = \mathbf{+}$. Flipping the sign gives $\lambda_u(J) < \lambda_u(MM^{\mathrm{T}})$. Furthermore, MM^{T} is at most rank two, so at least $n-2$ of its eigenvalues are zero. Therefore, at least $n-2$ eigenvalues of J are strictly less than zero. ■

By the above proposition and Lemma 4.2.1, only one eigenvalue has an undetermined sign. The next proposition provides a test to determine that sign.

Proposition 4.2.8 *Let $r \in R_{k,\omega,+}$ and $R := r^2$. Then generically r is stable if*

$$\xi(R) := 1 - \frac{1}{n}\sum_{u=1}^{n} \frac{\omega_u^2}{R s_u} > 0$$

and r is unstable if $\xi(R) < 0$.

Proof. For a generic $r \in R_{k,\omega,+}$, Lemmas 4.2.1 and 4.2.7 leave one unknown eigenvalue. Let λ_1 be this unknown value, $\lambda_2, \ldots, \lambda_{n-1}$ be the $n-2$ negative eigenvalues, and λ_n be the zero eigenvalue. Since $\lambda_n = 0$, the coefficient of the linear term in the characteristic equation for J is the product $-\lambda_1\lambda_2\cdots\lambda_{n-1}$. Thus we have

$$\begin{aligned}\det(J - \lambda I) &= \det(D + MM^{\mathrm{T}} - \lambda I)\\ &= \det(A + b\,b^{\mathrm{T}} + c\,c^{\mathrm{T}})\\ &= -(\lambda_1\lambda_2\cdots\lambda_{n-1})\lambda + O(\lambda^2)\end{aligned}$$

where $A := D - \lambda I$, b is the first column of M, and c is the second column of M. Applying Lemma 1.2.2 then gives

$$\det(A)\left[(1 + b^{\mathrm{T}}A^{-1}b)(1 + c^{\mathrm{T}}A^{-1}c) - (c^{\mathrm{T}}A^{-1}b)^2\right] = -(\lambda_1\lambda_2\cdots\lambda_{n-1})\lambda + O(\lambda^2) \tag{4.2.2}$$

We will consider each piece of the left-hand side individually. Again, note that for a generic $r \in R_{k,\omega,+}$ we have that $s_u > 0$.

1. $\det(A) = \det(D) + O(\lambda)$:

   We have that
$$\prod_{u=1}^{n}(-s_u - \lambda) = \left(\prod_{u=1}^{n} -s_u\right) + O(\lambda)$$

2. $A^{-1} = D^{-1} + \begin{bmatrix} s_1^{-2} & & \\ & \ddots & \\ & & s_n^{-2} \end{bmatrix}\lambda + O(\lambda^2)$:

   We have that
$$A^{-1} = \begin{bmatrix} \frac{-1}{s_1+\lambda} & & \\ & \ddots & \\ & & \frac{-1}{s_n+\lambda} \end{bmatrix}$$

   Using the Taylor series centered at zero on each diagonal element gives
$$\frac{-1}{s_u + \lambda} = -\frac{1}{s_u} + \frac{1}{s_u^2}\lambda + O(\lambda^2)$$

3. $1 + b^{\mathrm{T}} A^{-1} b = \dfrac{\lambda}{R} + O(\lambda^2)$:

   By Lemma 4.2.4, Proposition 4.2.6, and #2 we have

$$
\begin{aligned}
1 + b^{\mathrm{T}} A^{-1} b &= 1 + b^{\mathrm{T}} D^{-1} b + b^{\mathrm{T}} \begin{bmatrix} s_1^{-2} & & \\ & \ddots & \\ & & s_n^{-2} \end{bmatrix} b\lambda + O(\lambda^2) \\
&= 1 + \frac{1}{n} \sum_{u=1}^{n} \frac{-k_u^2 \operatorname{Re}^2(\phi_u)}{s_u} + \frac{1}{n} \sum_{u=1}^{n} \frac{k_u^2 \operatorname{Re}^2(\phi_u)}{s_u^2} \lambda + O(\lambda^2) \\
&= 1 - \frac{1}{n} \sum_{u=1}^{n} \frac{s_u}{R} + \frac{1}{n} \sum_{u=1}^{n} \frac{1}{R} \lambda + O(\lambda^2) \\
&= \frac{1}{R} f_{+}(R) + \frac{\lambda}{R} + O(\lambda^2) \\
&= \frac{\lambda}{R} + O(\lambda^2)
\end{aligned}
$$

   since $f_{+}(R) = 0$ by the assumption that $r \in R_{k,\omega,+}$.

4. $1 + c^{\mathrm{T}} A^{-1} c = \xi(R) + O(\lambda)$:

   Let $\xi(R) := 1 - \dfrac{1}{n} \displaystyle\sum_{u=1}^{n} \frac{\omega_u^2}{R s_u}$. By Lemma 4.2.4 and #2 we have

$$
\begin{aligned}
1 + c^{\mathrm{T}} A^{-1} c &= 1 + c^{\mathrm{T}} D^{-1} c + O(\lambda) \\
&= 1 + \frac{1}{n} \sum_{u=1}^{n} \frac{-k_u^2 \operatorname{Im}^2(\phi_u)}{s_u} + O(\lambda) \\
&= 1 - \frac{1}{n} \sum_{u=1}^{n} \frac{\omega_u^2}{R s_u} + O(\lambda) \\
&= \xi(R) + O(\lambda)
\end{aligned}
$$

5. $c^{\mathrm{T}} A^{-1} b = O(\lambda)$:

   By **IC1** (which is $\displaystyle\sum_{u=1}^{n} \omega_u = 0$), Lemma 4.2.4 and #2 we have

$$
\begin{aligned}
c^{\mathrm{T}} A^{-1} b &= c^{\mathrm{T}} D^{-1} b + O(\lambda) \\
&= \frac{1}{n} \sum_{u=1}^{n} \frac{-k_u^2 \operatorname{Re}(\phi_u) \operatorname{Im}(\phi_u)}{s_u} + O(\lambda) \\
&= \frac{-1}{n} \sum_{u=1}^{n} \frac{\omega_u}{R} + O(\lambda) \\
&= \frac{-1}{nR} \sum_{u=1}^{n} \omega_u + O(\lambda)
\end{aligned}
$$

$$= O(\lambda)$$

Putting #1–5 into the left-hand side of Eq. 4.2.2 gives

$$\begin{aligned}(\det(D)+O(\lambda))\left[\left(\frac{\lambda}{R}+O(\lambda^2)\right)(\xi(R)+O(\lambda))-(O(\lambda))^2\right] &= (\det(D)+O(\lambda))\left[\frac{\xi(R)}{R}\lambda+O(\lambda^2)\right] \\ &= \det(D)\frac{\xi(R)}{R}\lambda+O(\lambda^2)\end{aligned}$$

Thus Eq. 4.2.2 becomes

$$\det(D)\frac{\xi(R)}{R}\lambda+O(\lambda^2) = -(\lambda_1\lambda_2\cdots\lambda_{n-1})\lambda+O(\lambda^2)$$

Looking at the sign of the coefficient of λ gives

$$(-1)^n \operatorname{sign}(\xi(R)) = (-1)^{n-1}\operatorname{sign}(\lambda_1)$$

Hence $\operatorname{sign}(\lambda_1) = -\operatorname{sign}(\xi(R))$. Therefore if $\xi(R) > 0$, J will have $n-1$ negative eigenvalues, so r is stable, and if $\xi(R) < 0$, J will have a positive eigenvalue, so r is not stable. ■

In order to apply this test to the elements of $R_{k,\omega,+}$, we can rewrite ξ in terms of f_+ and its derivative. Since ξ is a factor of the linear coefficient of the characteristic equation which involves the derivatives of the KM equations, it is not surprising to see that ξ depends on f'. However, it is unexpected to see that ξ also depends on f and only on f and f'.

Lemma 4.2.9 $\xi(R) = \dfrac{f_+(R)}{R} - 2f_+'(R)$

Proof. Recall that $f_+(R) := -R + \dfrac{1}{n}\displaystyle\sum_{u=1}^{n} s_u$ where $s_u := \sqrt{k_u^2 R - \omega_u^2}$. Thus

$$f_+'(R) = -1 + \frac{1}{n}\sum_{u=1}^{n}\frac{k_u^2}{2s_u}$$

Therefore, we have that

$$\begin{aligned}\frac{f_+(R)}{R} - 2f_+'(R) &= -1 + \frac{1}{n}\sum_{u=1}^{n}\frac{s_u}{R} + 2 - \frac{1}{n}\sum_{u=1}^{n}\frac{k_u^2}{s_u} \\ &= 1 - \frac{1}{n}\sum_{u=1}^{n}\left(\frac{k_u^2}{s_u} - \frac{s_u}{R}\right) \\ &= 1 - \frac{1}{n}\sum_{u=1}^{n}\frac{k_u^2 R - s_u^2}{R s_u} \\ &= 1 - \frac{1}{n}\sum_{u=1}^{n}\frac{\omega_u^2}{R s_u} \\ &= \xi(R)\end{aligned}$$

■

Since all of the stable equilibria can be found within $R_{k,\omega,+}$ by Proposition 4.2.6, it is necessary to know when that set is nonempty in order to prove the existence of a stable solution.

Lemma 4.2.10 *If Eq. 4.0.1 has at least one equilibrium, then $R_{k,\omega,+} \neq \emptyset$*

Proof. Suppose that Eq. 4.0.1 has at least one equilibrium. By Theorem 4.1.6, we have that $R_{k,\omega,\sigma} \neq \emptyset$ for some $\sigma \in \{-,+\}^n$. Recall that by construction, $f_\sigma(r^2) = 0$ for any $r \in R_{k,\omega,\sigma}$. Thus we have that $f_+(r^2) \geq 0$. Moreover, $\lim_{R\to\infty} f_+(R) = -\infty$. Therefore by the Intermediate Value Theorem, f_+ must have a root, so $R_{k,\omega,+} \neq \emptyset$. ■

Combining all the previous results, we can now prove the main stability result for the rank one coupling case.

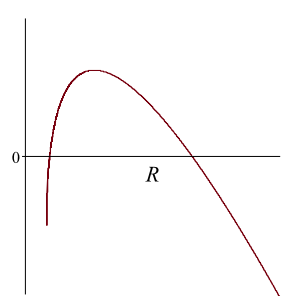
Theorem 4.2.11 *Suppose that Eq. 4.0.1 has at least one equilibrium. Then generically Eq. 4.0.1 has a unique stable equilibrium. Furthermore, this stable equilibrium, if it exists, comes from the largest root of f_+.*

Proof. By Lemma 4.2.10, we have that $R_{k,\omega,+} \neq \emptyset$ if Eq. 4.0.1 has at least one equilibrium. Also by Proposition 4.2.6, $R_{k,\omega,+}$ contains all the stable equilibria. Thus we need to consider each element of $R_{k,\omega,+}$, which by construction are the roots of f_+. We have that

$$f_+''(R) = -\frac{1}{n}\sum_{u=1}^{n} \frac{k_u^4}{4s_u^3} < 0$$

Thus f_+ is concave down everywhere on its domain, so it has at most two roots. Supposing that f_+ has at least one root, there are then generically two cases to consider.

- f_+ has two roots, R_1 and R_2:

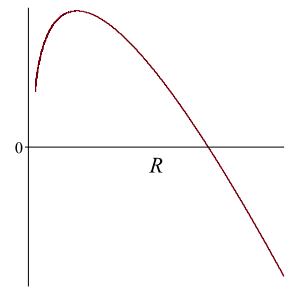


  Let $R_1 < R_2$. By Lemma 4.2.9,

$$\xi(R_u) = \frac{f_+(R_u)}{R_u} - 2f_+'(R_u) = -2f_+'(R_u)$$

  Since $f_+(R)$ is concave down, we have that $f_+'(R_1) > 0$ and $f_+'(R_2) < 0$. So by Proposition 4.2.8, R_1 is unstable and R_2 is stable.

- f_+ has one root with multiplicity one $\hat{R}$:



We must have that $f_+'(\hat{R}) < 0$, so again by Lemma 4.2.9 and Proposition 4.2.8, $\hat{R}$ is stable.

■

4.3 Counting

As summarized in Section 1.2.4, an upper bound on the number of equilibria to Eq. 4.0.1 is $2, 6, 14$ for $n = 2, 3, 4$, respectively. Theorem 4.3.3 shows that $2^n - 2$ bounds the number of equilibria with Corollary 4.3.4 showing that $2^n - 2$ is actually the generic root count for the polynomial system version of Eq. 4.0.1 modulo shift.

Let $\omega \in \mathbb{R}^n$ and $k \in \mathbb{R}^n_{>0}$ satisfy **IC1**-**IC3**. The following shows that the function

$$g(R) \;:=\; \prod_{\sigma \in \{-,+\}^n} f_\sigma(R) \;=\; \prod_{\sigma \in \{-1,+1\}^n} \left(-R + \frac{1}{n} \sum_{u=1}^{n} \sigma_u \sqrt{k_u^2 R - \omega_u^2} \right) \tag{4.3.1}$$

is actually a reducible polynomial.

Proposition 4.3.1 *The univariate function g in Eq. 4.3.1 is a polynomial of degree 2^n. Moreover, there exists a polynomial $h(R)$ of degree $2^n - 2$ with*

$$g(R) = R^2 \cdot h(R)$$

Proof. Since g is a product over all 2^n conjugates, it immediately follows that g is a polynomial with leading term $(-R)^{2^n}$ showing that g is a polynomial of degree 2^n.

In order to show that R^2 is a factor of g, we simply need to show that $g(0) = g'(0) = 0$. To that end, consider $\sigma_\omega = \text{sign}(\omega)$. (We will let $\text{sign}(\omega_u) = +$ if $\omega_u = 0$.) Then,

$$f_{\sigma_\omega}(0) = \frac{1}{n} \sum_{u=1}^{n} \text{sign}(\omega_u) \sqrt{-\omega_u^2} \;=\; \frac{\sqrt{-1}}{n} \sum_{u=1}^{n} \omega_u \;= 0$$

by **IC1**. By Eq. 4.3.1, this immediately shows that $g(0) = 0$ since one of the terms in the product is 0.

By a similar argument as above, $f_{-\sigma_\omega}(0)=0$ by **IC1**. This shows that at least two terms in the product defining g in Eq. 4.3.1 are zero. Hence, the product rule for differentiation shows that $g'(0)=0$. ■

Example 4.3.2 *For $n=2$, we have*

$$g(R)=R^4-\frac{1}{2}\left(k_1^2+k_2^2\right)R^3+\frac{1}{16}\left(\left(k_1^2-k_2^2\right)^2+8\left(\omega_1^2+\omega_2^2\right)\right)R^2$$
$$-\frac{1}{8}\left(k_1^2-k_2^2\right)\left(\omega_1^2-\omega_2^2\right)R+\frac{1}{16}\left(\omega_1^2-\omega_2^2\right)^2$$

which is indeed a polynomial of degree $2^2=4$. Moreover, **IC1** *implies $\omega_2=-\omega_1$ so that*

$$g(R)=R^2\left(R^2-\frac{1}{2}\left(k_1^2+k_2^2\right)R+\frac{1}{16}\left(\left(k_1^2-k_2^2\right)^2+16\omega_1^2\right)\right)$$

Proposition 4.3.1 immediately provides the following upper bound.

Theorem 4.3.3 *If $\omega\in\mathbb{R}^n$ and $k\in\mathbb{R}^n_{>0}$ satisfy* **IC1–IC3**, *then there are at most 2^n-2 equilibria satisfying Eq. 4.0.1.*

Proof. This follows from Theorem 4.1.6 since $g(R)$ in Eq. 4.3.1 has at most 2^n-2 positive roots. ■

Corollary 4.3.4 *The generic root count modulo shift to the polynomial version of Eq. 4.0.1 is 2^n-2.*

Proof. Reviewing the proof of Theorem 4.1.6 shows that 2^n-2 also bounds the number of complex solutions to Eq. 4.0.1. For g in Eq. 4.3.1, $g''(0)\neq 0$ for generic values of the parameters yielding that there are generically 2^n-2 nonzero roots of g. Hence, 2^n-2 is the generic root count of Eq. 4.0.1. ■

Example 4.3.5 *Table 4.1 shows that the polynomial system of Eq. 4.0.1 for $n=4$, $\omega=\left(-\frac{3}{4},-\frac{3}{4},\frac{1}{4},\frac{3}{4}\right)$, and $k=(\sqrt{1.5},\sqrt{1.5},\sqrt{1.5},\sqrt{1.5})$ has 12 complex roots modulo shift, which is less than the generic root count of $2^4-2=14$. In fact, as in the proof of Proposition 4.3.1, this is due to the following four quantities being equal to zero:*

$$\sum_{u=1}^{4}\omega_u,\quad \sum_{u=1}^{4}-\omega_i,\quad \omega_1-\omega_2-\omega_3+\omega_4,\quad -\omega_1+\omega_2+\omega_3-\omega_4$$

Hence, g in Eq. 4.3.1 has $g(0)=g'(0)=g''(0)=g'''(0)=0$, namely

$$g(R)=\frac{R^4}{1073741824}(64R^4-96R^3+20R^2+1)(64R^2-24R+9)^2(64R^2-24R+1)^2$$

Theorem 4.3.3 provides an upper bound of 2^n-2 when the symmetric coupling matrix has rank one while [2] provides an upper bound of $\binom{2n-2}{n-1}$ in the general case. By Stirling's formula,

$$\binom{2n-2}{n-1}\approx\frac{4^{n-1}}{\sqrt{\pi(n-1)}}$$

showing the bound in Theorem 4.3.3 for the rank-one case is roughly the square root of the general purpose bound from [2].

Counting equilibria for particular cases

Motivated by [50], we use Theorem 4.1.6 to analyze the number of equilibria satisfying Eq. 4.0.1 for particular cases when n is even (Proposition 4.3.6 and Corollary 4.3.9) and when n is odd (Proposition 4.3.12).

Proposition 4.3.6 *Suppose that $n \geq 2$ is even and $q > 0$. For $\omega = \left(nq, \ldots, nq, -nq, \ldots, -nq\right)$ and $k = (n, \ldots, n)$, there are exactly*

$$2^n - \sum_{-q<\ell<q} \binom{n}{n/2+\ell}$$

equilibria satisfying Eq. 4.0.2 counting multiplicity. Hence, the number of equilibria changes precisely at the integers $q = 1, 2, \ldots, n/2$.

Proof. Since $k_u^2 = n^2$ and $\omega_u^2 = n^2 q^2$, Theorem 4.1.6 shows that we need to compute all $R > 0$ where

$$\begin{aligned} R &= \frac{1}{n} \sum_{u=1}^{n} \sigma_u \sqrt{n^2 R - n^2 q^2} \\ &= \sum_{u=1}^{n} \sigma_u \sqrt{R - q^2} \\ &= S \sqrt{R - q^2} \end{aligned} \tag{4.3.2}$$

with $S := \sum_{u=1}^{n} \sigma_u$ and $\sigma \in \{-,+\}^n$.

If $S \leq 0$, then Eq. 4.3.2 has no positive solutions. Since n is even, the remaining cases have $S \geq 2$. Thus, the positive solutions of Eq. 4.3.2 must satisfy

$$R = \frac{S}{2}\left(S \pm \sqrt{S^2 - 4q^2}\right) > 0$$

This yields three cases:

- $2 \leq S < 2q$: Eq. 4.3.2 has no positive solutions;
- $S = 2q \geq 2$: Eq. 4.3.2 has one positive solution of multiplicity 2, namely $R = S^2/2$;
- $S > 2q$ with $S \geq 2$: Eq. 4.3.2 has two distinct positive solutions.

Suppose that q is not an integer. Since S is even, we have $S \neq 2q$. Hence, the number of equilibria is exactly

$$2 \cdot \#\left\{\sigma \in \{+,-\}^n : S > 2q\right\} = 2 \cdot \#\left\{\sigma \in \{+,-\}^n : S \geq 2\lceil q \rceil\right\} = 2 \cdot \sum_{\ell=\lceil q \rceil}^{n/2} \binom{n}{n/2+\ell}$$

Since $\binom{n}{n/2+\ell} = \binom{n}{n/2-\ell}$ and $2^n = \sum_{\ell=0}^{n} \binom{n}{\ell}$, the number of equilibria when q is not an integer is

$$2 \cdot \sum_{\ell=\lceil q \rceil}^{n/2} \binom{n}{n/2+\ell} = \sum_{\ell=-n/2}^{-\lceil q \rceil} \binom{n}{n/2+\ell} + \sum_{\ell=\lceil q \rceil}^{n/2} \binom{n}{n/2+\ell} = 2^n - \sum_{-q<\ell<q} \binom{n}{n/2+\ell}$$

When q is an integer, we need to add in the case when $S = 2q$ yielding

$$2 \cdot \#\{\sigma \in \{+,-\}^n : S \geq 2q\} = 2 \cdot \sum_{\ell=q}^{n/2} \binom{n}{n/2+\ell} = 2^n - \sum_{-q<\ell<q} \binom{n}{n/2+\ell}$$

■

Example 4.3.7 *For $n = 2$ and $q > 0$, the case of $\omega = (2q, -2q)$ and $k = (2,2)$ corresponds to $\omega = \left(\frac{q}{2}, -\frac{q}{2}\right)$ and $k = (1,1)$. Hence, counting multiplicity, there are two equilibria for $q \leq 1$ and no equilibria for $q > 1$ in agreement with Proposition 4.3.6.*

Example 4.3.8 *For $n = 4$ and $q > 0$, the case of $\omega = (4q, 4q, -4q, -4q)$ and $k = (4,4,4,4)$ corresponds to $\omega = (q/4, q/4, -q/4, -q/4)$ and $k = (1,1,1,1)$. Figure 4.21(a) plots the regions based on the number of equilibria when $k = (1,1,1,1)$ such that $\omega_3 = \omega_4 = -(\omega_1 + \omega_2)/2$. With this setup, $\omega_1 = \omega_2 = q/4$ implies $\omega_3 = \omega_4 = -q/4$. Since the sign is arbitrary, the plot in Figure 4.21(b) incorporates the line $\omega_1 = \omega_2 = q/4$. By Proposition 4.3.6, there are 10 equilibria for $0 < |q| < 1$, 2 equilibria for $1 < |q| < 2$, and no equilibria for $|q| > 2$.*

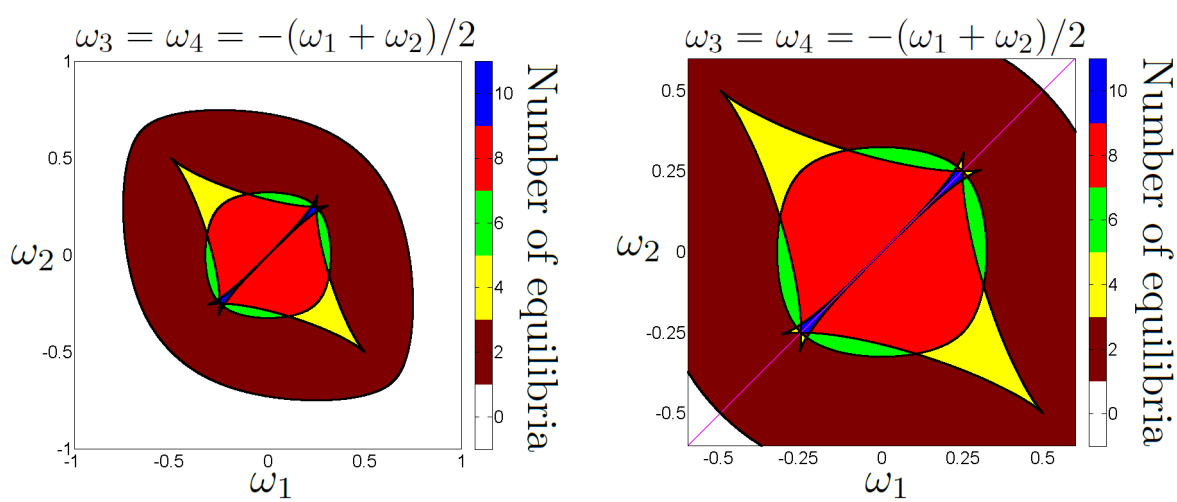


Figure 4.21 Regions based on the number of equilibria satisfying Eq. 4.0.1 for $n = 4$ with a restricted set of ω and $k = (1,1,1,1)$. The diagonal line in (b) corresponds to results from Proposition 4.3.6.

Proposition 4.3.6 immediately yields the following.

Corollary 4.3.9 *Suppose that $n \geq 2$ is even and $q > 0$. The maximum number of distinct equilibria satisfying* (4.0.1) *when $\omega = (nq, \ldots, nq, -nq, \ldots, -nq)$ and $k = (n, \ldots, n)$ is*

$$2^n - \binom{n}{n/2} \tag{4.3.3}$$

which occurs for all $0 < q < 1$.

Example 4.3.10 *For $n = 4$, Corollary 4.3.9 provides a maximum of $2^4 - \binom{4}{2} = 10$ distinct equilibria which matches the computational results in [50].*

Before considering the odd case, we first define the constants

$$q_o := \frac{\sqrt{414 - 66\sqrt{33}}}{16} \approx 0.3690 \quad \text{and} \quad R_o := \frac{21 - 3\sqrt{33}}{8} \approx 0.4708 \tag{4.3.4}$$

and prove an inequality regarding them.

Lemma 4.3.11 *For $0 < q < q_o$, $R_o + \sqrt{R_o} - 2\sqrt{R_o - q^2} < 0$ where q_o and R_o as defined in Eq. 4.3.4.*

Proof. Since $q < q_o$ and $R_o - q^2 > R_o - q_o^2 > 0$, we have

$$R_o + \sqrt{R_o} - 2\sqrt{R_o - q^2} \; < \; R_o + \sqrt{R_o} - 2\sqrt{R_o - q_o^2} \; = 0$$

■

With Lemma 4.3.11, we now consider the case when n is odd.

Proposition 4.3.12 *Suppose that $n \geq 3$ is odd and let $0 < q < q_o$ where q_o is defined by* (4.3.4)*. For $\omega = (nq, \ldots, nq, -nq, \ldots, -nq, 0)$ and $k = (n, \ldots, n)$, the number of equilibria satisfying Eq. 4.0.1 is*

$$2^n - \binom{n-1}{(n-1)/2} \tag{4.3.5}$$

Proof. Since $k_u^2 = n^2$ for $u = 1, 2, \ldots, n$, $\omega_v^2 = n^2 q^2$ for $v = 1, 2, \ldots, n-1$, and $\omega_n = 0$, Theorem 4.1.6 shows that we need to compute all $R > 0$ with

$$\begin{aligned} R &= \frac{1}{n} \sum_{u=1}^{n} \left(\sigma_u \sqrt{n^2 R - n^2 q^2} \right) + \frac{1}{n} \sigma_n \sqrt{n^2 R} \\ &= \sum_{u=1}^{n-1} \left(\sigma_u \sqrt{R - q^2} \right) + \sigma_n \sqrt{R} \\ &= S \sqrt{R - q^2} + \sigma_n \sqrt{R} \end{aligned} \tag{4.3.6}$$

where $S := \sum_{u=1}^{n} \sigma_u$ and $\sigma \in \{-,+\}^n$. Define $p_\sigma(R) := R - \sigma_n\sqrt{R} - S\sqrt{R-q^2}$. Since $n-1$ is even, we know that S is also even. This yields three cases to consider.

- $S < 0$:

  Rewriting Eq. 4.3.6 as
  $$R - \sigma_n\sqrt{R} = S\sqrt{R-q^2}$$
  shows that the right-hand size is non-positive. Hence, to have a solution, we need $\sigma_n = +$ and $R \in (q^2, 1)$. Since $p_\sigma(q^2) = q^2 - q < 0$ and $p_\sigma(1) = -S\sqrt{1-q^2} > 0$, we know that there is at least one root in $(q^2, 1)$. In fact, since $S \leq -2$, it is easy to see that p_σ is a strictly increasing function on $(q^2, 1)$ since
  $$p'_\sigma(R) = 1 - \frac{1}{2\sqrt{R}} + \frac{-S}{2\sqrt{R-q^2}} \geq 1 - \frac{1}{2\sqrt{R}} + \frac{1}{\sqrt{R-q^2}} \geq 1 + \frac{1}{2\sqrt{R}} > 0$$
  for all $R \in (q^2, 1)$. Thus, this case yields one equilibrium for each $\sigma \in \{-,+\}^n$ such that $\sigma_n = +$ and $S < 0$ for a total of
  $$\frac{1}{2}\left(2^{n-1} - \binom{n-1}{(n-1)/2}\right)$$

- $S = 0$:

  Since (4.3.6) becomes $R = \sigma_n\sqrt{R}$, this case requires $\sigma_n = +$ and $R = 1$. The total number of equilibria for this case is thus
  $$\binom{n-1}{(n-1)/2}$$

- $S > 0$:

  We split this into two cases based on the value of σ_n.

  - $\sigma_n = +$:

    Rewriting Eq. 4.3.6 as
    $$R - \sqrt{R} = S\sqrt{R-q^2}$$
    shows that the right-hand size is nonnegative. Hence, to have a solution, we need $R > 1$. Since $p_\sigma(1) = -S\sqrt{1-q^2} < 0$ and $\lim_{R\to\infty} p_\sigma(R) = \infty$, we know that there is at least one root in $(1, \infty)$. In fact, the root is unique since the graph of p_σ is concave up due to
    $$p''_\sigma(R) = \frac{1}{4R^{3/2}} + \frac{S}{4(R-q^2)^{3/2}} > 0$$
    for $R > 1$. Hence, the total number of equilibria for this case is
    $$\frac{1}{2}\left(2^{n-1} - \binom{n-1}{(n-1)/2}\right)$$

- $\sigma_n = -$:

  We need to compute the number of roots of p_σ for $R > q^2$. Since $S \geq 2$ and $R^{3/2} > (R-q^2)^{3/2}$ for all $R > q^2$, it follows that

$$p''_\sigma(R) = -\frac{1}{4R^{3/2}} + \frac{S}{4(R-q^2)^{3/2}} > 0$$

  when $R > q^2$. Hence, p_σ is concave up on $R > q^2$ with $p_\sigma(q^2) = q^2 + q > 0$ and $\lim_{R\to\infty} p_\sigma(R) = \infty$. Thus, the number of roots depends on the sign of the minimum value of p_σ on $R > q^2$. Since increasing S makes p_σ more negative and Lemma 4.3.11 shows that $p_\sigma(R_o) < 0$ when $S = 2$, there are always two roots with $R > q^2$. Hence, the total number of equilibria for this case is

$$2^{n-1} - \binom{n-1}{(n-1)/2}$$

The result is obtained by simply summing the number of equilibria from all of these cases. ■

Example 4.3.13 *For $n = 3$, Proposition 4.3.12 shows that the number of equilibria for $\omega = (3q, -3q, 0)$ and $k = (3,3,3)$ is $2^3 - \binom{2}{1} = 6$ whenever $0 < q < q_o$ with q_o defined in* (4.3.4)*. This is equivalent to the case when $\omega = (q/3, -q/3, 0)$ and $k = (1,1,1)$ for $0 < q < q_o$. Since the ordering of the elements in ω is arbitrary, Figure 4.22 is an enhanced version of Figure 1.3 that plots, in red, the corresponding three segments within the region having* 6 *equilibria:*

- $\{(\alpha, 0, -\alpha) : |\alpha| < q_0\}$ *is the horizontal segment,*
- $\{(0, \alpha, -\alpha) : |\alpha| < q_0\}$ *is the vertical segment, and*
- $\{(\alpha, -\alpha, 0) : |\alpha| < q_0\}$ *is the diagonal segment.*

The following suggests an upper bound on the maximum number of equilibria.

Conjecture 4.3.14 *For $n \geq 2$, the maximum number of equilibria satisfying Eq. 4.0.1 with n oscillators is*

$$\begin{cases} 2^n - \binom{n}{n/2} & \text{if } n \text{ is even} \\ 2^n - \binom{n-1}{(n-1)/2} & \text{if } n \text{ is odd} \end{cases}$$

which are achieved in Corollary 4.3.9 and Proposition 4.3.12, respectively.

This conjecture matches the known cases of $n = 2$ and $n = 3$, and agrees with the conjecture for $n = 4$ provided in [50] for the standard Kuramoto model.

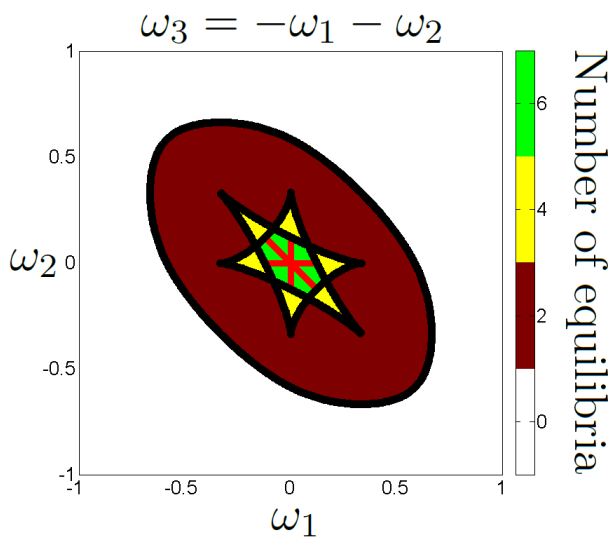


Figure 4.22 Enhanced version of Figure 1.3 with the three segments from Example 4.3.13 plotted in red

Asymptotic behavior

Even though we can only conjecture an upper bound on the number of equilibria, the results from Corollary 4.3.9 and Proposition 4.3.12 provide the following result: there can asymptotically be as many equilibria satisfying Eq. 4.0.1 as the number of complex solutions to Eq. 4.0.1 modulo shift.

Proposition 4.3.15 *As $n \to \infty$, the ratio of the maximum number of equilibria satisfying Eq. 4.0.1 and the generic root count to Eq. 4.0.1 limits to* 1.

Proof. For each $n \geq 2$, let $\Omega(n)$ denote this ratio. Theorem 4.3.3 and Proposition 4.3.12 together with Corollaries 4.3.4 and 4.3.9 show that, for every $\ell \geq 1$,

$$\frac{2^{2\ell} - \binom{2\ell}{\ell}}{2^{2\ell} - 2} \leq \Omega(2\ell) \leq 1 \quad \text{and} \quad \frac{2^{2\ell+1} - \binom{2\ell}{\ell}}{2^{2\ell+1} - 2} \leq \Omega(2\ell+1) \leq 1$$

Stirling's formula yields

$$\lim_{\ell\to\infty} \frac{\binom{2\ell}{\ell}}{2^{2\ell} - 2} = \lim_{\ell\to\infty} \frac{\frac{2^{2\ell}}{\sqrt{\pi\ell}}}{2^{2\ell} - 2} = 0$$

so that

$$\begin{aligned}
1 &\geq \lim_{\ell\to\infty} \Omega(2\ell) \\
&\geq \lim_{\ell\to\infty} \frac{2^{2\ell} - \binom{2\ell}{\ell}}{2^{2\ell} - 2} \\
&= \lim_{\ell\to\infty} \frac{2^{2\ell}}{2^{2\ell} - 2} - \lim_{\ell\to\infty} \frac{\binom{2\ell}{\ell}}{2^{2\ell} - 2} \\
&= 1 - 0 = 1
\end{aligned}$$

Similarly, Stirling's formula yields

$$\lim_{\ell\to\infty} \frac{\binom{2\ell}{\ell}}{2^{2\ell+1}-2} = \lim_{\ell\to\infty} \frac{\frac{2^{2\ell}}{\sqrt{\pi\ell}}}{2^{2\ell+1}-2} = 0$$

so that

$$\begin{aligned}
1 &\geq \lim_{\ell\to\infty} \Omega(2\ell+1) \\
&\geq \lim_{\ell\to\infty} \frac{2^{2\ell+1}-\binom{2\ell}{\ell}}{2^{2\ell+1}-2} \\
&= \lim_{\ell\to\infty} \frac{2^{2\ell+1}}{2^{2\ell+1}-2} - \lim_{\ell\to\infty} \frac{\binom{2\ell}{\ell}}{2^{2\ell+1}-2} \\
&= 1-0=1
\end{aligned}$$

Therefore, $\Omega(n) \to 1$ as $n \to \infty$. ■

CHAPTER

5

ARBITRARY RANK COUPLING

The Kuramoto model with arbitrary rank coupling has the form

$$\frac{\mathrm{d}\theta_u}{\mathrm{d}t} = \omega_u - \frac{1}{n}\sum_{v=1}^{n} K_{uv}\sin(\theta_u - \theta_v) \qquad \text{for } u = 1, 2, \ldots, n \tag{5.0.1}$$

where $n \geq 2$ is the number of oscillators, $K \in \mathbb{R}^{n\times n}$ is symmetric and gives the coupling strengths, θ is the phase angles of the oscillators, and ω is the natural frequencies of the oscillators. As discussed in Section 1.1.1, setting $\frac{\mathrm{d}\theta_u}{\mathrm{d}t} = 0$ gives the equation for the equilibria when using the rotating reference frame. Thus the equations to solve for finding the equilibria can be simplified to

$$\omega_u = \frac{1}{n}\sum_{v=1}^{n} K_{uv}\sin(\theta_u - \theta_v) \qquad \text{for } u = 1, 2, \ldots, n \tag{5.0.2}$$

We will take the following input conditions.

IC1: $\sum_{u=1}^{n} \omega_u = 0$

This is necessary to use the rotating reference frame.

IC2: $\omega_u \neq 0$ for $u = 1, 2, \ldots, n$

We will avoid cases where any oscillator has an intrinsic velocity equal to the average.

5.1 Finding

Note that if θ is a solution to Eq. 3.0.1, then $\theta + c$ is also a solution for any constant c. Thus, two solutions are called equivalent modulo shift if each component-wise difference is the same modulo 2π. In order to choose a representative, an output condition for $\theta \in (-\pi, \pi]^n$ is given below so that only one of these equivalent solutions is chosen. To set this up, a few variables will be defined. Let $L, R^{\mathrm{T}} \in \mathbb{R}^{n\times\rho}$ where $\rho := \mathrm{rk}(K)$ be factors of K. In other words, $LR = K$. Let $\phi_u := e^{i\theta_u}$ and let $\psi := R\phi$. Now we state the output condition as follows.

OC: $\quad \exists q \;\; \psi_q \in \mathbb{R}_+ \text{ and } \psi_{q+1} = \cdots = \psi_\rho = 0.$

Remark 5.1.1 ***OC*** *always produces a unique representative as long as* ***IC2*** *is satisfied which is shown in the proof of Lemma 5.1.4.*

Definition 5.1.2 *Let $\Theta_{K,\omega}$ be the set of all solutions to Eq. 5.0.1 satisfying* **OC***, that is,*

$$\Theta_{K,\omega} := \left\{ \theta \in (-\pi, \pi]^n : \underset{u}{\forall} \;\; \omega_u = \frac{1}{n}\sum_{v=1}^{n} K_{uv} \sin(\theta_u - \theta_v) \wedge \mathbf{OC} \right\}$$

Let $\mathbb{U}$ be the unit circle in the complex plane, $\mathbb{U} := \{z \in \mathbb{C} : \; |z| = 1\}$. We can perform a series of transformations on $\Theta_{K,\omega}$ to transform it into a system of radical equations in potentially less variables if the rank of K is relatively small compared to n.

Lemma 5.1.3 *Let*

$$f : \left([\theta_1, \ldots, \theta_n]^{\mathrm{T}}\right) \longmapsto \left[e^{i\theta_1}, \ldots, e^{i\theta_n}\right]^{\mathrm{T}}$$

Then we have

1. *$f|_{\Theta_{K,\omega}}$ is injective.*
2. *$f\left(\Theta_{K,\omega}\right) = \Phi_{C,\omega}$ where*

$$\Phi_{K,\omega} := \left\{ \phi \in \mathbb{U}^n : \underset{u}{\forall} \; n\omega_u = \mathrm{Im}\left(\phi_u r_u^*\right) \wedge \mathbf{OC} \right\}$$

and where

$$r := K\phi.$$

(Note that r_u^* denotes the complex conjugate of r_u.)

Proof.

1. Let $\theta, \theta' \in \Theta_{K,\omega}$ be such that $f(\theta) = f(\theta')$. Then for every u we have $e^{i\theta_u} = e^{i\theta'_u}$, and thus $\theta_u = \theta'_u$ since $\theta_u, \theta'_u \in (-\pi, \pi]$. Hence $\theta = \theta'$.

2. Let $r := K\phi$. Note that z^* denotes the complex conjugate of z. Then

$$\begin{aligned}
f\left(\Theta_{K,\omega}\right) &= \left\{ f(\theta) : \underset{u}{\forall}\ \omega_u = \frac{1}{n}\sum_{v=1}^{n} K_{uv}\sin(\theta_u - \theta_v) \wedge \mathbf{OC} \right\} \\
&= \left\{ f(\theta) : \underset{u}{\forall}\ n\omega_u = \sum_{v=1}^{n} K_{uv}\,\mathrm{Im}\left(e^{i(\theta_u-\theta_v)}\right) \wedge \mathbf{OC} \right\} \\
&= \left\{ \phi \in \mathbb{U}^n : \underset{u}{\forall}\ n\omega_u = \mathrm{Im}\left(\sum_{v=1}^{n} K_{uv}\phi_u\phi_v^*\right) \wedge \mathbf{OC} \right\} \\
&= \left\{ \phi \in \mathbb{U}^n : \underset{u}{\forall}\ n\omega_u = \mathrm{Im}\left(\phi_u \sum_{v=1}^{n} K_{uv}\phi_v^*\right) \wedge \mathbf{OC} \right\} \\
&= \left\{ \phi \in \mathbb{U}^n : \underset{u}{\forall}\ n\omega_u = \mathrm{Im}\left(\phi_u r_u^*\right) \wedge \mathbf{OC} \right\}
\end{aligned}$$

■

Lemma 5.1.4 *We have*

$$\Phi_{K,\omega} = \bigcup_{\sigma\in\{-,+\}^n} \Phi_{K,\omega,\sigma}$$

where

$$\Phi_{K,\omega,\sigma} := \left\{ \phi \in \mathbb{U}^n : \underset{u}{\forall}\ \phi_u = \frac{n\omega_u i + \sigma_u\sqrt{|r_u|^2 - n^2\omega_u^2}}{r_u^*} \wedge \mathbf{OC} \right\}$$

and $r := K\phi$.

Proof. We have that

$$\begin{aligned}
\Phi_{K,\omega} &= \left\{ \phi \in \mathbb{U}^n : \underset{u}{\forall}\ n\omega_u = \mathrm{Im}\left(\phi_u r_u^*\right) \wedge \mathbf{OC} \right\} \\
&= \left\{ \phi \in \mathbb{U}^n : \underset{u}{\forall}\ 2n\omega_u i = \phi_u r_u^* - \phi_u^* r_u \wedge \mathbf{OC} \right\}
\end{aligned}$$

Note that $|\phi_u| = 1$, so $\phi_u \neq 0$. Thus multiplying through by ϕ_u gives us that

$$\begin{aligned}
\Phi_{K,\omega} &= \left\{ \phi \in \mathbb{U}^n : \underset{u}{\forall}\ 2n\omega_u i\phi_u = r_u^*\phi_u^2 - r_u \wedge \mathbf{OC} \right\} \\
&= \left\{ \phi \in \mathbb{U}^n : \underset{u}{\forall}\ 0 = r_u^*\phi_u^2 - 2n\omega_u i\phi_u - r_u \wedge \mathbf{OC} \right\}
\end{aligned}$$

For $\phi \in \Phi_{K,\omega}$, we have that $r_u^* \neq 0$. Otherwise, we would have $0 = r_u^*\phi_u^2 - 2n\omega_u i\phi_u + r_u = -2n\omega_u i\phi_u$ and thus either $\phi_u = 0$ or $\omega_u = 0$, a contradiction. Therefore we can use the quadratic formula to solve for ϕ_u.

$$\Phi_{K,\omega} = \bigcup_{\sigma\in\{-,+\}^n} \left\{ \phi \in \mathbb{U}^n : \underset{u}{\forall}\ \phi_u = \frac{2n\omega_u i + \sigma_u\sqrt{-4n^2\omega_u^2 + 4r_u^* r_u}}{2r_u^*} \wedge \mathbf{OC} \right\}$$

$$= \bigcup_{\sigma\in\{-,+\}^n} \left\{ \phi \in \mathbb{U}^n : \underset{u}{\forall}\ \phi_u = \frac{n\omega_u i + \sigma_u \sqrt{|r_u|^2 - n^2\omega_u^2}}{r_u^*} \wedge \mathbf{OC} \right\}$$

■

Remark 5.1.5 *Note that*

$$n\omega_u = \mathrm{Im}(\phi_u r_u^*) = \mathrm{Im}\left(n\omega_u i + \sigma_u \sqrt{|r_u|^2 - n^2\omega_u^2} \right)$$

Therefore,

$$|r_u|^2 \geq n^2\omega_u^2$$

Lemma 5.1.6 *Let*

$$g : \phi \longmapsto R\phi.$$

Then we have that

1. $g|_{\Phi_{K,\omega,\sigma}}$ *is injective.*
2. $g(\Phi_{K,\omega,\sigma}) = \Psi_{K,\omega,\sigma}$ *where*

$$\Psi_{K,\omega,\sigma} := \left\{ \psi \in \mathbb{C}^\rho : \underset{u}{\forall}\ \psi_u = \sum_{v=1}^{n} R_{uv} \frac{n\omega_u i + \sigma_v \sqrt{\left|\hat{r}_v\right|^2 - n^2\omega_u^2}}{\hat{r}_v^*} \wedge \mathbf{OC} \right\}$$

and $\hat{r} := L\psi$.

Proof.

1. Let $\phi, \phi' \in \Phi_{K,\omega,\sigma}$ be such that $g(\phi) = g(\phi')$. Then

$$r = K\phi = LR\phi = Lg\left(\phi\right) = Lg(\phi') = LR\phi' = K\phi' = r'$$

Thus

$$\phi_u = \frac{n\omega_u i + \sigma_u \sqrt{|r_u|^2 - n^2\omega_u^2}}{r_u^*} = \frac{n\omega_u i + \sigma_u \sqrt{|r_u'|^2 - n^2\omega_u^2}}{\left(r_u'\right)^*} = \phi_u'$$

2. Note that $g(\phi) = \psi$. Recall that $r := K\phi$ and define $\hat{r} := L\psi$. We have that

$$\begin{aligned}
g(\Phi_{K,\omega,\sigma}) &= \left\{ g(\phi) : \underset{u}{\forall}\ \phi_u = \frac{n\omega_u i + \sigma_u \sqrt{|r_u|^2 - n^2\omega_u^2}}{r_u^*} \wedge \mathbf{OC} \right\} \\
&= \left\{ \psi \in \mathbb{C}^\rho : \psi = R\phi \wedge \underset{u}{\forall}\ \phi_u = \frac{n\omega_u i + \sigma_u \sqrt{|r_u|^2 - n^2\omega_u^2}}{r_u^*} \wedge \mathbf{OC} \right\} \\
&= \left\{ \psi \in \mathbb{C}^\rho : \underset{u}{\forall}\ \psi_u = \sum_{v=1}^{n} R_{uv}\phi_v \wedge \underset{u}{\forall}\ \phi_u = \frac{n\omega_u i + \sigma_u \sqrt{|r_u|^2 - n^2\omega_u^2}}{r_u^*} \wedge \mathbf{OC} \right\}
\end{aligned}$$

$$
\begin{aligned}
&= \left\{ \psi \in \mathbb{C}^{\rho} : \underset{u}{\forall}\ \psi_u = \sum_{v=1}^{n} R_{uv} \left(\frac{n\omega_v i + \sigma_v \sqrt{|\gamma_v|^2 - n^2\omega_v^2}}{r_v^*} \right) \wedge \right. \\
&\qquad \left. \underset{u}{\forall}\ \phi_u = \frac{n\omega_u i + \sigma_u \sqrt{|\gamma_u|^2 - n^2\omega_u^2}}{r^*} \wedge \mathbf{OC} \right\} \\
&= \left\{ \psi \in \mathbb{C}^{\rho} : \underset{u}{\forall}\ \psi_u = \sum_{v=1}^{n} R_{uv} \left(\frac{n\omega_v i + \sigma_v \sqrt{|\hat{\gamma}_v|^2 - n^2\omega_v^2}}{\hat{r}_v^*} \right) \wedge \mathbf{OC} \right\}
\end{aligned}
$$

since

$$r = K\phi = LR\phi = L\psi = \hat{r}$$

■

Combining the above Lemmas gives a reformulation for $\Theta_{K,\omega}$.

Theorem 5.1.7 *We have*

$$\Theta_{K,\omega} = F\left(\bigcup_{\sigma \in \{-,+\}^n} G_\sigma\left(\Psi_{K,\omega,\sigma}\right) \right)$$

where

$$
\begin{aligned}
F\left(\phi\right) &= \left[\arg\left(\phi_1\right), \ldots, \arg\left(\phi_n\right)\right]^{\mathrm{T}} \\
G_\sigma\left(\psi\right) &= \left[\frac{n\omega_1 i + \sigma_1 \sqrt{\left|\hat{r}_1\right|^2 - n^2\omega_1^2}}{\hat{r}_1^*}, \ldots, \frac{n\omega_n i + \sigma_n \sqrt{\left|\hat{r}_n\right|^2 - n^2\omega_n^2}}{\hat{r}_n^*} \right]^{\mathrm{T}}
\end{aligned}
$$

and $\hat{r} := L\psi$. Furthermore, F and G_σ are injective.

Proof. We split the proof into three parts.

1. $F = f^{-1}$ and is injective.

   Recall that

   $$f\left([\theta_1, \ldots, \theta_n]^{\mathrm{T}}\right) = \left[e^{i\theta_1}, \ldots, e^{i\theta_n}\right]^{\mathrm{T}}$$

   Hence it is immediate that

   $$f^{-1}\left(\phi\right) = \left[\arg\left(\phi_1\right), \ldots, \arg\left(\phi_n\right)\right]^{\mathrm{T}} = F\left(\phi\right)$$

   and that F is injective.

2. $G_\sigma = g^{-1}|_{\Phi_{K,\omega,\sigma}}$ and is injective.

   Recall that

   $$g\left(\phi\right) = R\phi.$$

Let $g(\phi) = \psi$ where $\phi \in \Phi_{K,\omega,\sigma}$. Then

$$\gamma = K\phi = LR\phi = L\psi = \hat{r},$$

so

$$\phi_u = \frac{n\omega_u i + \sigma_u \sqrt{|r_u|^2 - n^2\omega_u^2}}{r_u^*} = \frac{n\omega_u i + \sigma_u \sqrt{|\hat{r}_u|^2 - n^2\omega_u^2}}{\hat{r}_u^*} = G_\sigma(\psi)_u.$$

Furthermore, let $\psi, \psi' \in \Psi_{K,\omega,\sigma}$ be such that $G_\sigma(\psi) = G_\sigma(\psi')$. Then

$$\hat{r} = L\psi = LRG_\sigma(\psi) = LRG_\sigma(\psi') = L\psi' = \hat{r}'$$

Thus

$$\psi_u = \sum_{v=1}^{n} R_{uv} \frac{n\omega_u i + \sigma_v \sqrt{|\hat{r}_v|^2 - n^2\omega_u^2}}{\hat{r}_v^*} = \sum_{v=1}^{n} R_{uv} \frac{n\omega_u i + \sigma_v \sqrt{|\hat{r}_v'|^2 - n^2\omega_u^2}}{(\hat{r}_v')^*} = \psi_u'$$

3. $\Theta_{K,\omega} = F\left(\bigcup_{\sigma\in\{-,+\}^n} G_\sigma(\Psi_{K,\omega,\sigma})\right)$.

Combining Lemmas 5.1.3, 5.1.4, and 5.1.6 with parts one and two above, we have

$$\Theta_{K,\omega} = f^{-1}(\Phi_{K,\omega}) = f^{-1}\left(\bigcup_{\sigma\in\{-,+\}^n} \Phi_{K,\omega,\sigma}\right) = f^{-1}\left(\bigcup_{\sigma\in\{-,+\}^n} g^{-1}(\Psi_{K,\omega,\sigma})\right)$$

■

In order to make use of the reformulation theorem and turn it into an algorithm, we need to split up the equilibria in $\Psi_{K,\omega,\sigma}$ based on q in **OC**.

Definition 5.1.8 *Let*

$$\Psi^q_{K,\omega,\sigma} := \left\{ \psi \in \mathbb{C}^\rho : \forall_u \left(\psi_u = \sum_{v=1}^{n} R_{uv} \frac{n\omega_u i + \sigma_v \sqrt{|\hat{r}_v|^2 - n^2\omega_u^2}}{\hat{r}_v^*} \right) \wedge \psi_q > 0 \wedge \psi_{q+1} = \cdots = \psi_\rho = 0 \right\}$$

Note that $\Psi_{K,\omega,\sigma} = \bigcup_{q\in\{1,2,\ldots,\rho\}} \Psi^q_{K,\omega,\sigma}$.

Furthermore, we can convert $\Psi^q_{K,\omega,\sigma}$ to use equations over the real numbers by breaking the equations for ψ_u into real and imaginary components. This allows for solvers that function over only the real numbers to be used to find $\Psi_{K,\omega,\sigma}$. Moreover, we can bound the values of the real and imaginary components by the absolute row sums of R.

Proposition 5.1.9 *Let* $x := \mathrm{Re}(\psi)$ *and* $y := \mathrm{Im}(\psi)$. *Then*

$$\begin{aligned}\Psi^q_{K,\omega,\sigma} = \Big\{ x + y\,i :\ & x, y \in \mathbb{R}^\rho \wedge \\ & x = -R(SLy + TLx) \wedge \\ & y = \ \ R(SLx + TLy) \wedge \\ & \underset{u>q}{\forall}\ x_u = y_u = 0 \wedge \\ & x_q > 0,\ y_q = 0 \Big\}\end{aligned}$$

where

$$S := \begin{bmatrix} \frac{n\omega_1}{(Lx)_1^2+(Ly)_1^2} & & \\ & \ddots & \\ & & \frac{n\omega_n}{(Lx)_1^2+(Ly)_n^2} \end{bmatrix} \qquad T := \begin{bmatrix} \frac{\sqrt{(Lx)_1^2+(Ly)_1^2-n^2\omega_1^2}}{(Lx)_1^2+(Ly)_1^2} & & \\ & \ddots & \\ & & \frac{\sqrt{(Lx)_n^2+(Ly)_n^2-n^2\omega_n^2}}{(Lx)_n^2+(Ly)_n^2} \end{bmatrix}$$

Proof. Let $\psi \in \Psi_{K,\omega,\sigma}$. Then we have that

$$\begin{aligned} x_u &= \mathrm{Re}(\psi_u) \\ &= \mathrm{Re}\left(\sum_{v=1}^n R_{uv} \frac{n\omega_v i + \sigma_v \sqrt{|\hat{r}_v|^2 - n^2\omega_v^2}}{\hat{r}_v^*}\right) \\ &= \mathrm{Re}\left(\sum_{v=1}^n R_{uv}\, \hat{r}_u \frac{n\omega_v i + \sigma_v \sqrt{|\hat{r}_v|^2 - n^2\omega_v^2}}{|\hat{r}_v^2|}\right) \\ &= \sum_{v=1}^n \frac{R_{uv}}{|\hat{r}_v^2|} \mathrm{Re}\Big(n\omega_v \hat{r}_v i + \sigma_v \hat{r}_v \sqrt{|\hat{r}_v|^2 - n^2\omega_v^2}\Big) \end{aligned}$$

Recall that

$$|\hat{r}_u|^2 \geq n^2\omega_u^2$$

for any equilibria (see Remark 5.1.5). Thus

$$\begin{aligned} x_u &= \sum_{v=1}^n \frac{R_{uv}}{|\hat{r}_v|^2} \mathrm{Re}\Big(n\omega_v \hat{r}_v i + \sigma_v \hat{r}_v \sqrt{|\hat{r}_v|^2 - n^2\omega_v^2}\Big) \\ &= \sum_{v=1}^n \frac{R_{uv}}{|\hat{r}_v|^2} \Big(-n\omega_v \mathrm{Im}(\hat{r}_v) + \sigma_v \mathrm{Re}(\hat{r}_v) \sqrt{|\hat{r}_v|^2 - n^2\omega_v^2}\Big) \end{aligned}$$

Doing the same for the imaginary parts gives the following.

$$y_u = \mathrm{Im}(\phi_u)$$

$$
\begin{aligned}
&= \operatorname{Im}\left(\sum_{v=1}^{n} R_{uv} \frac{n\omega_v i + \sigma_v \sqrt{|\hat{r}_v|^2 - n^2\omega_v^2}}{\hat{r}_v^*}\right) \\
&= \operatorname{Im}\left(\sum_{v=1}^{n} R_{uv} \hat{r}_u \frac{n\omega_v i + \sigma_v \sqrt{|\hat{r}_v|^2 - n^2\omega_v^2}}{|\hat{r}_v|^2}\right) \\
&= \sum_{v=1}^{n} \frac{R_{uv}}{|\hat{r}_v|^2} \operatorname{Im}\left(n\omega_v \hat{r}_v i + \sigma_v \hat{r}_v \sqrt{|\hat{r}_v|^2 - n^2\omega_v^2}\right) \\
&= \sum_{v=1}^{n} \frac{R_{uv}}{|\hat{r}_v|^2} \operatorname{Im}\left(n\omega_v \hat{r}_v i + \sigma_v \hat{r}_v \sqrt{|\hat{r}_v|^2 - n^2\omega_v^2}\right) \\
&= \sum_{v=1}^{n} \frac{R_{uv}}{|\hat{r}_v|^2} \left(n\omega_v \operatorname{Re}(\hat{r}_v) + \sigma_v \operatorname{Im}(\hat{r}_v) \sqrt{|\hat{r}_v|^2 - n^2\omega_v^2}\right)
\end{aligned}
$$

Also, we have that

$$\hat{r} = L\psi = L(x + y\,i) = (Lx) + (Ly)i$$

Therefore, $\operatorname{Re}(\hat{r}) = Lx$, $\operatorname{Im}(\hat{r}) = Ly$, and

$$|\hat{r}_u|^2 = (Lx)_u^2 + (Ly)_u^2$$

Thus we now have

$$
\begin{aligned}
x_u &= \sum_{v=1}^{n} \frac{R_{uv}}{(Lx)_v^2 + (Ly)_v^2} \left(-n\omega_v (Ly)_v + \sigma_v (Lx)_v \sqrt{(Lx)_v^2 + (Ly)_v^2 - n^2\omega_v^2}\right) \\
y_u &= \sum_{v=1}^{n} \frac{R_{uv}}{(Lx)_v^2 + (Ly)_v^2} \left(n\omega_v (Lx)_v + \sigma_v (Ly)_v \sqrt{(Lx)_v^2 + (Ly)_v^2 - n^2\omega_v^2}\right)
\end{aligned}
$$

Rearranging the equations gives

$$
\begin{aligned}
x_u &= \sum_{v=1}^{n} R_{uv} \left(-S_v (Ly)_v + T_v (Lx)_v\right) \\
y_u &= \sum_{v=1}^{n} R_{uv} \left(S_v (Lx)_v + T_v (Ly)_v\right)
\end{aligned}
$$

where

$$S_v := \frac{n\omega_v}{(Lx)_v^2 + (Ly)_v^2} \qquad\qquad T_v := \frac{\sigma_v \sqrt{(Lx)_v^2 + (Ly)_v^2 - n^2\omega_v^2}}{(Lx)_v^2 + (Ly)_v^2}$$

By letting $S := \operatorname{diag}(S_1, \ldots, S_n)$ and $T := \operatorname{diag}(T_1, \ldots, T_n)$, we can combine the equations for x and for y using matrix multiplication which gives the result. ■

Proposition 5.1.10 *Let $x + y\,i \in \Psi^q_{K,\omega,\sigma}$ where $x, y \in \mathbb{R}^n$. Then $|x_u|,\ |y_u| \le \sum_{v=1}^{n} |R_{uv}|$ for all u.*

Proof. Let $\psi \in \Psi_{K,\omega,\sigma}$ and let $x := \mathrm{Re}(\psi)$ and $y := \mathrm{Im}(\psi)$. Let $\phi := G_\sigma(\psi)$ and recall that $\phi \in \mathbb{U}^n$. Then we have that

$$|x_u|, |y_u| \leq |\psi_u| = \left|\sum_{v=1}^{n} R_{uv}\phi_v\right| \leq \sum_{v=1}^{n} |R_{uv}\phi_v| = \sum_{v=1}^{n} |R_{uv}||\phi_v| \leq \sum_{v=1}^{n} |R_{uv}|$$

■

Putting all of this together gives an algorithm to find $\Theta_{K,\omega}$. The following algorithm relies on three outside functions. **rk**(M) returns the rank of the matrix M; **Factor**(M) returns two matrices $L, R^{\mathrm{T}} \in \mathbb{C}^{n\times\rho}$ where ρ is the rank of M such that $LR = M$; and **Solve**$(\Psi^q_{K,\omega,\sigma}, B)$ returns all the solutions satisfying the set of equations in $\Psi^q_{K,\omega,\sigma}$ (i.e., the elements of that set) within the bounding box B.

Algorithm 5.1.11

In: $K \in \mathbb{R}^{n\times n}$ *a symmetric matrix and* $\omega \in \mathbb{R}^n$ *satisfying* **IC1–IC2**.

Out: $\Theta_{K,\omega}$

1. $\rho \leftarrow$ **rk**(K)
2. $L, R \leftarrow$ **Factor**(K)
3. $\Theta_{K,\omega} \leftarrow \{\}$
4. *For* $q \in \{1, \ldots, \rho\}$
   (a) $B \leftarrow$ *bounds for* x *and* y *using Proposition 5.1.10*
   (b) *For* $\sigma \in \{-,+\}^n$
      i. $S \leftarrow$ **Solve**$\left(\Psi^q_{K,\omega,\sigma}, B\right)$
      ii. $\Theta_{K,\omega} \leftarrow \Theta_{K,\omega} \cup F(G_\sigma(S))$
5. *Return* $\Theta_{K,\omega}$

Note that Algorithm 5.1.11 calls a system solve $\rho 2^n$ times. Thus it is unlikely to offer any performance benefits compared to other methods of finding all the roots of Eq. 5.0.1 unless perhaps ρ is very small compared to n. However, the reformulation could be very advantageous when searching for only the stable equilibria as will be shown in the next section.

5.2 Stability

In this section we will show that by taking the restriction that $K \succ 0$, all of the stable solutions (in the orbitally stable sense; see Section 1.1.2 for a discussion of orbital stability) can be found from

the all positive sign case. Let J be the Jacobian for Eq. 5.0.1. A straightforward calculation gives

$$J = -\frac{1}{n}\begin{bmatrix} \sum_{v=1}^{n} K_{1v}\cos(\theta_1-\theta_v) & & \\ & \ddots & \\ & & \sum_{v=1}^{n} K_{nv}\cos(\theta_n-\theta_v) \end{bmatrix} + \frac{1}{n}\Big[K_{uv}\cos(\theta_u-\theta_v)\Big]_{(u,v)} \tag{5.2.1}$$

Lemma 5.2.1 *The Jacobian matrix has a zero eigenvalue.*

Proof. It is immediate from Eq. 5.2.1 that the sum of all the columns of J is zero. Therefore the all ones vector is an eigenvector with the corresponding eigenvalue of zero. ■

The all ones eigenvector corresponds to the invariance under shift of the Kuramoto model. Since the model depends only on the differences between angles, we will ignore the shift when determining the stability of an equilibrium. Moreover, note that J is a real symmetric matrix, so all its eigenvalues are real. Thus if J evaluated at an equilibrium has $n-1$ negative eigenvalues, then that equilibrium is stable. Likewise, if J has a positive eigenvalue, then that equilibrium cannot be stable [9].

It is helpful for the main result of this section to rewrite J in terms of the variables used in the reformulation.

Proposition 5.2.2 *Let $\phi_u := e^{i\theta_u}$ and $r := K\phi$. Then for a given $\sigma \in \{-,+\}^n$ we have that $J = D + \frac{1}{n}K \circ MM^{\mathrm{T}}$ where*

$$D := -\frac{1}{n}\begin{bmatrix} \sigma_1\sqrt{|r_1|^2 - n^2\omega_1^2} & & \\ & \ddots & \\ & & \sigma_n\sqrt{|r_n|^2 - n^2\omega_n^2} \end{bmatrix}$$

$$M := \begin{bmatrix} \mathrm{Re}(\phi_1) & \mathrm{Im}(\phi_1) \\ \vdots & \vdots \\ \mathrm{Re}(\phi_n) & \mathrm{Im}(\phi_n) \end{bmatrix} \tag{5.2.2}$$

(Note that $\circ$ is the Hadamard Product from Definition 1.2.3.)

Proof. Following the same procedure as Lemma 5.1.3, we have that

$$\begin{aligned} \sum_{v=1}^{n} K_{uv}\cos(\theta_u-\theta_v) &= \sum_{v=1}^{n} K_{uv}\,\mathrm{Re}\big(e^{i(\theta_u-\theta_v)}\big) \\ &= \mathrm{Re}\left(\phi_u \sum_{v=1}^{n} K_{uv}\phi_v^*\right) \\ &= \mathrm{Re}\big(\phi_u r_u^*\big) \end{aligned}$$

Therefore we can rewrite the Jacobian as

$$J = -\frac{1}{n}\begin{bmatrix} \mathrm{Re}\left(\phi_1 r_1^*\right) & & \\ & \ddots & \\ & & \mathrm{Re}\left(\phi_n r_n^*\right) \end{bmatrix} + \frac{1}{n}\left[K_{uv}\,\mathrm{Re}\left(\phi_u \phi_v^*\right)\right]_{(u,v)}$$

For an equilibrium ψ, we have that

$$G_\sigma(\psi) = \phi_u = \frac{n\omega_u i + \sigma_u\sqrt{|r_u|^2 - n^2\omega_u^2}}{r_u^*}$$

from Theorem 5.1.7. Therefore, we have that $\mathrm{Re}\left(\phi_u r_u^*\right) = \sigma_u\sqrt{|r_u|^2 - n^2\omega_u^2}$ giving

$$J = -\frac{1}{n}\begin{bmatrix} \sigma_1\sqrt{|r_1|^2 - n^2\omega_1^2} & & \\ & \ddots & \\ & & \sigma_n\sqrt{|r_n|^2 - n^2\omega_n^2} \end{bmatrix} + \frac{1}{n}K \circ \left[\mathrm{Re}\left(\phi_u \phi_v^*\right)\right]_{(u,v)}$$

Furthermore,

$$\left[\mathrm{Re}\left(\phi_u \phi_v^*\right)\right]_{(u,v)} = \begin{bmatrix} \mathrm{Re}(\phi_1) & \mathrm{Im}(\phi_1) \\ \vdots & \vdots \\ \mathrm{Re}(\phi_n) & \mathrm{Im}(\phi_n) \end{bmatrix}\begin{bmatrix} \mathrm{Re}(\phi_1) & \cdots & \mathrm{Re}(\phi_n) \\ \mathrm{Im}(\phi_1) & \cdots & \mathrm{Im}(\phi_n) \end{bmatrix}$$

■

Theorem 5.2.3 *If $K \succeq 0$, then the set of stable solutions to Eq. 5.0.1 is a subset of $F\left(G_{\mathbf{+}}\left(\Psi_{K,\omega,\mathbf{+}}\right)\right)$.*

Proof. If $K \succeq 0$, then the Schur Product Theorem (Theorem 1.2.4) gives

$$K \circ MM^{\mathrm{T}} \succeq 0$$

Furthermore, by Lemma 1.2.1 we have

$$\lambda_u(J) = \lambda_u\left(D + \frac{1}{n}K \circ MM^{\mathrm{T}}\right) \geq \lambda_u(D)$$

Therefore, if $-\sigma_u\sqrt{|\hat{r}_u|^2 - n^2\omega_u^2} > 0$ for any u, then J will have a positive eigenvalue. Hence for any equilibria, if $\sqrt{|\hat{r}_u|^2 - n^2\omega_u^2} > 0$, we must have $\sigma_u = +$, and if $\sqrt{|\hat{r}_u|^2 - n^2\omega_u^2} = 0$, then σ_u can be $+$ or $-$. Thus the associated root for a stable solution is an element of $\Psi_{K,\omega,\mathbf{+}}$. ■

Using this theorem, we can replace the for loop in line $4b$ of Algorithm 5.1.11 with $\sigma \leftarrow \mathbf{+}$ to find a superset of the stable solutions. Thus the solver would get called only ρ times which could offer a significant performance advantage when ρ is relatively small compared to n. Prototype Matlab code is provide in the Appendix for such a situation.

BIBLIOGRAPHY

[1] D. Aeyels and J. A. Rogge, *Existence of Partial Entrainment and Stability of Phase Locking Behavior of Coupled Oscillators*, Progress of Theoretical Physics, 112 (2004), pp. 921–942.

[2] J. Baillieul and C. Byrnes, *Geometric critical point analysis of lossless power system models*, IEEE Trans. Circu. Syst., 29 (1982), pp. 724–737.

[3] K. Bar-Eli, *On the stability of coupled chemical oscillators*, Physica D Nonlinear Phenomena, 14 (1985), pp. 242–252.

[4] D. J. Bates, J. D. Hauenstein, A. J. Sommese, and C. W. Wampler, *Bertini: Software for numerical algebraic geometry*, Available at www.nd.edu/∼sommese/bertini.

[5] R. Bellman, *Introduction to Matrix Analysis (2nd Ed.)*, Society for Industrial and Applied Mathematics, USA, 1997.

[6] J .C. Bronski, L. DeVille, and M. J. Park, *Fully synchronous solutions and the synchronization phase transition for the finite-N Kuramoto model*, Chaos, 22 (2012), 033133.

[7] B. Buchberger, *Ein Algorithmus zum Auffinden der Basiselemente des Restklassenringes nach einem nulldimensionalen Polynomideal [An Algorithm for Finding the Basis Elements in the Residue Class Ring Modulo a Zero Dimensional Polynomial Ideal]*, (Trans. in Journal of Symbolic Comp., Special Issue on Logic, Math., and Comp. Science: Interactions, 41 (2006) pp. 475–511.) Mathematical Institute, University of Innsbruck, Austria, 1965.

[8] J. Canny and I. Emiris, *An Efficient Algorithm for the Sparse Mixed Resultant*, Proc. 10th Intern. Symp. on Applied Algebra, Algebraic Algorithms, and Error-Correcting Codes, Lect. Notes in Comp. Science, 263 (1993), pp. 89–104.

[9] J. Carr, *Applications of Centre Manifold Theory*, Springer, New York, 1982.

[10] Z. Charles and A. Zachariah, *Efficiently finding all power flow solutions to tree networks*, 55th Annu. Allerton Conf. Commun., Control, Comput., Oct. 3 – Oct. 5, 2017.

[11] T. Chen and D. Mehta, *On the Network Topology Dependent Solution Count of the Algebraic Load Flow Equations*, IEEE Transactions on Power Systems, 33 (2018), pp. 1451–1460.

[12] T. Chen, D. Mehta, and M. Niemerg, *A network topology dependent upper bound on the number of equilibria of the Kuramoto model*, arXiv:1603.05905, 2016.

[13] T. Chen, R. Davis, and D. Mehta, *Counting Equilibria of the Kuramoto Model Using Birationally Invariant Intersection Index*, SIAM J. Appl. Algebra Geom, 2 (2018), pp. 489–507.

[14] Y. Choi, S. Ha, S. Jung, and Y. Kim, *Asymptotic formation and orbital stability of phase-locked states for the Kuramoto model*, Physica D: Nonlinear Phenomena, 241 (2012), pp. 735–754.

[15] O. Coss, J. D. Hauenstein, H. Hong, and D. K. Molzahn, *Locating and Counting Equilibria of the Kuramoto Model with Rank-One Coupling*, SIAM J. Appl. Algebra Geom, 2 (2018), pp. 45–71.

[16] D. Cox, J. Little, and D. O'Shea. *Using Algebraic Geometry*, Springer-Verlag, New York, 2005.

[17] D. Cox, J. Little, and D. O'Shea. *Ideals, Varieties, and Algorithms*, Springer International Publishing, 2015.

[18] R. Delabays, P. Jacquod, and F. Dörfler, *The Kuramoto Model on Oriented and Signed Graphs*, SIAM Journal on Applied Dynamical Systems, 18 (2019), pp. 458–480.

[19] F. Dörfler and F. Bullo, *On the Critical Coupling for Kuramoto Oscillators*, SIAM Journal on Applied Dynamical Systems, 10 (2011), pp. 1070–1099.

[20] F. Dörfler and F. Bullo, *Synchronization and transient stability in power networks and nonuniform Kuramoto oscillators*, SIAM Journal on Control and Optimization, 50 (2012), pp. 1616–1642.

[21] F. Dörfler, M. Chertkov, and F. Bullo, *Synchronization in complex oscillator networks and smart grids*, Proceedings of the National Academy of Sciences, 110 (2013), pp. 2005–2010.

[22] G. B. Ermentrout, *An adaptive model for synchrony in the firefly* Pteroptyx malaccae, Journal of Mathematical Biology, 29 (1991), pp. 571–585.

[23] D. R. Grayson, and M. E. Stillman, *Macaulay2, a software system for research in algebraic geometry*, Available at http://www.math.uiuc.edu/Macaulay2/.

[24] R. Hammer, M. Hocks, U. Kulisch and D. Ratz, *C++ Toolbox for Verified Computing I: Basic Numerical Problems Theory, Algorithms, and Programs*, Springer-Verlag, Berlin, 1995.

[25] J. D. Hauenstein, A. J. Sommese, and C. W. Wampler, *Regeneration homotopies for solving systems of polynomials*, Mathematics of Computation, 80 (2011), pp. 345–377.

[26] I. A. Hiskens and R. J. Davy, *Exploring the power flow solution space boundary*, IEEE Trans. Power Systems, 16 (2001) pp. 389–395.

[27] Y. Ikeda, *An Interacting Agent Model of Economic Crisis*, arXiv:2001.11843, 2020.

[28] A. Jadbabaie, N. Motee, and M. Barahona, *On the Stability of the Kuramoto Model of Coupled Nonlinear Oscillators*, Proceedings of the American Control Conference, Boston, MA, June 2004.

[29] A. Korsak, *On the Question of Uniqueness of Stable Load-Flow Solutions*, IEEE Transactions of Power Apparatus and Systems, PAS-91 (1972), pp. 1093–1100.

[30] W. Krämer, *C-XSC: A powerful environment for reliable computations in the natural and engineering sciences*, 2011 4th International Conference on Biomedical Engineering and Informatics (BMEI), Shanghai, 2011, pp. 2130–2134. Software documentation available at http://www2.math.uni-wuppertal.de/wrswt/xsc/cxsc/apidoc/html/index.html.

[31] Y. Kuramoto, *Self-entrainment of a population of coupled non-linear oscillators*, International Symposium on Mathematical Problems in Theoretical Physics: January 23–29, 1975, Kyoto University, Kyoto/Japan, Springer, Berlin, 1975, pp. 420–422.

[32] Y. Kuramoto, *Chemical Oscillations, Waves, and Turbulence*, Springer, Berlin, 1984.

[33] B. C. Lesieutre and D. Wu, *An efficient method to locate all the load flow solutions – revisited*, 53rd Annu. Allerton Conf. Commun., Control, Comput., Sept. 29 – Oct. 2, 2015.

[34] Z. Lu, K. Klein-Cardeña, S. Lee, T. M. Antonsen, M. Girvan, and E. Ott, *Resynchronization of circadian oscillators and the east-west asymmetry of jet-lag*, Chaos, 26 (2016), 094811.

[35] W. Ma and S. Thorp, *An efficient algorithm to locate all the load flow solutions*, IEEE Trans. Power Syst., 8 (1993), pp. 1077–1083.

[36] F. Macaulay, *Some Formulae in elimination*, Proc. London. Math. Soc., 33 (1902), pp. 3–27.

[37] Y. Maistrenko, O. Popovych, and P. Tass, *Chaotic Attractor in the Kuramoto Model*, International Journal of Bifurcation and Chaos, 15 (2015), pp. 3457–3466.

[38] D. Mehta, N. S. Daleo, F. Dörfler, and J. D. Hauenstein, *Algebraic geometrization of the Kuramoto model: equilibria and stability analysis*, Chaos, 25 (2015), 053103.

[39] D. Mehta, H. Nguyen and K. Turitsyn, *Numerical Polynomial Homotyopy Continuation Method to Locate All The Power Flow Solutions*, IET Generation, Transmission & Distribution, 10 (2016), pp. 2972–2980.

[40] D. K. Molzahn, B. C. Lesieutre, and H. Chen, *Counterexample to a continuation-based algorithm for finding all power flow solutions*, IEEE Trans. Power Syst., 28 (2013), pp. 564–565.

[41] D. K. Molzahn, M. Niemerg, D. Mehta, and J. D. Hauenstein, *Investigating the Maximum Number of Real Solutions to the Power Flow Equations: Analysis of Lossless Four-Bus Systems*, arXiv:1603.05908, 2016.

[42] A. P. Morgan and A. J. Sommese, *Coefficient-parameter polynomial continuation*, Appl. Math. Comput., 29 (1989), pp. 123–160.

[43] J. C. Neu, *The method of near-identity transformations and its applications*, SIAM Journal on Applied Mathematics, 38 (1980), pp. 189–208.

[44] J. Pantaleone, *Synchronization of Metronomes*, American Journal of Physics, 70 (2002), pp. 992–1000.

[45] F. M. A. Salam, L. Ni, S. Guo, and X. Sun, *Parallel processing for the load flow of power systems: the approach and applications*, IEEE 28th Ann. Conf. Decis. Control (CDC), 1989, pp. 2173–2178.

[46] I. Schur, *Bemerkungen zur Theorie der beschränkten Bilinearformen mit unendlich vielen Veränderlichen*, Journal für die reine und angewandte Mathematik, 140 (1911), pp. 1–28.

[47] H. Sompolinsky, D. Golomb, and D. Kleinfeld, *Global processing of visual stimuli in a neural network of coupled oscillators*, Proceedings of the National Academy of Sciences, 87 (1990), pp. 7200–7204.

[48] B. Sturmfels, *Sparse elimination theory*, Proc. Comput. Algebr. Geom. Commut. Algebra, D. Eisenbud, L. Robbiano, (Eds.), 1991.

[49] K. Wiesenfeld, P. Colet, and S. H. Strogatz, *Frequency locking in Josephson arrays: connection with the Kuramoto model*, Phys. Rev. E, 57 (1998), pp. 1563–1569.

[50] X. Xin, T. Kikkawa, and Y. Liu, *Analytical solutions of equilibrium points of the standard Kuramoto model: 3 and 4 oscillators*, American Control Conf. (ACC), 2016, pp. 2447–2452.

APPENDICES

APPENDIX

A

CODE EXAMPLES

A.1 Rank One Solver

The following C++ code implements Algorithms 4.1.21 and 4.1.25 simultaneously using the C-XSC library to find the positive roots of f_σ [30]. This library uses an interval Newton method to find real roots to a specified precision [24]. Since the left hand side of the domain of f_σ has an undefined derivative value, a few special considerations have to be made to accommodate this solving method. Namely, a tolerance value is used as the buffer from the left hand end of the domain, and the left edge up to the buffer is checked manually. The input is assumed to satisfy **IC1 – 3**, but is checked to determine it **IC4** is satisfied to decide which algorithm to use. Note that C++ uses zero based indexing, so all the index values are decreased by one compared to the statements in Section 4.1. Also note that this implementation cannot handle cases with $n > 64$ since the largest integer type in C++ is 64 bits.

```
#include <nlfzero.hpp> // C-XSC library nonlinear equations solver
#include <stacksz.hpp> // Increase stack size for some special C++ compilers
#include <cmath>       // For abs() and sqrt()

using namespace cxsc;
using namespace std;

// Global variables
int n, prune_num;
unsigned long long j;
real *w_sqrs, *k_sqrs, inner_min;
```

```
12
13 // Univariate radical equation f_sigma
14 DerivType f_sigma(const DerivType &R){
15     int u, sgn;
16     DerivType inner, f = -R;
17
18     for (u = 0; u < n; u++){
19         // Conver the sign number j's postion u bit into + or -
20         sgn = ((j >> (n-1-u)) & 1)*2 - 1;
21
22         // Force the value inside the radical to be above the min value by
23         // shifting to the right as needed
24         // This avoids issues where the floating point error range extends to
25         // where f or f' is undefined which crashes the solver
26         inner = k_sqrs[u]*R - w_sqrs[u];
27         if (Inf(fValue(inner)) < inner_min){
28             inner = inner - Inf(fValue(inner)) + inner_min;
29         }
30         f = f + sgn*sqrt(inner)/n;
31     }
32     return f;
33 }
34
35 // The pruning function
36 unsigned long long P(unsigned long long sgn_num){
37     int u, zeros_found = 0;
38
39     // Search for zeros from the right
40     for (u = 0; u < n; u++){
41         if (((sgn_num >> u) & 1) == 0){
42             zeros_found++;
43             if (zeros_found == prune_num){
44                 // Zero out everything from this zero onward
45                 sgn_num = ~((1 << u)-1) & sgn_num;
46
47                 // Subtract one unless the number is already zero
48                 if (sgn_num > 0)
49                     sgn_num--;
50                 return sgn_num;
51             }
52         }
53     }
54     // Less than prune_num zeros, so return 0
55     return 0;
56 }
57
58
59 int main (){
```

```
    int u, v, sgn, num_roots, error;
    unsigned long long int num_given_up_intervals=0;
    real *w, *k;
    real k_sqr_min, pos_sgn_sum, tolerance, lhs, rhs, theta;
    interval value, I;
    ivector zeros;
    intvector unique;
    bool flag;

    const double pi = 3.14159265358979323846264338327 95;

    // Set the output format
    cout << SetPrecision(23,15) << Scientific;

    // Get the number of parameters (Note that 2 <= n <= 64 is assumed)
    cout << "Number of parameters: ";
    cin >> n;

    // Set up storage for the parameters and output
    w = new real[n];
    k = new real[n];
    w_sqrs = new real[n];  // Global so f_sigma can access
    k_sqrs = new real[n];  // Global so f_sigma can access

    // Get the parameters
    // (Note that IC1-3 are assumed to be satisfied)
    cout << "w: ";
    for (u = 0; u < n; u++){
        cin >> w[u];
    }

    cout << "k: ";
    for (u = 0; u < n; u++){
        cin >> k[u];
    }

    // Store the squares to avoid recomputing them
    // Also find the min k^2 value
    k_sqr_min = sqr(k[0]);
    for (u = 0; u < n; u++){
        w_sqrs[u] = sqr(w[u]);
        k_sqrs[u] = sqr(k[u]);
        if (k_sqrs[u] < k_sqr_min)
            k_sqr_min = k_sqrs[u];
    }

    // Set the left side of the solution intervals
    lhs = w_sqrs[n-1]/k_sqrs[n-1];
```

```
108
109     // Check if IC4 is satisfied to determine the pruning level
110     prune_num = 2;
111     for (u = 0; u < n-1; u++){
112         if (k[u] < k[u+1]){
113             prune_num = 1;
114             break;
115         }
116     }
117
118     // Get a tolerance value
119     cout << "Tolerance: ";
120     cin >> tolerance;
121     cout << endl;
122
123     // Compute the starting sign number
124     if (n == 64)
125         j = ~0;
126     else
127         j = ((unsigned long long)1 << n) - 1;
128
129     // Iterate over the valid sign cases
130     while (j > 0){
131         // Determine the rhs of the interval
132         pos_sgn_sum = 0;
133         for (u = 0; u < n; u++){
134             pos_sgn_sum += ((int)((j >> u) & 1))*k[n-1-u];
135         }
136         rhs = sqr(pos_sgn_sum/n);
137
138         // If the interval is empty use the pruning function to skip cases
139         if (rhs < lhs){
140             j = P(j);
141             continue;
142         }
143
144         // We cannot use the root finding method since the f_sigma' is
145         // undefined at the left endpoint of our interval
146         // So instead we will check if a superset of the range of f_sigma on
147         // the interval [lhs, lhs+tolerance] contains zero
148         // If it does not, we can safely ignore this subinterval
149         // Otherwise, we record that there is an interval on which we cannot
150         // determine if there are solutions.
151         flag = true;
152         fEval(f_sigma, interval(lhs, lhs+tolerance), value);
153         if (Inf(value) <= 0 && Sup(value) >= 0){
154             num_given_up_intervals++;
155             flag = false;
```

```
        }

        // Make sure the interval is still valid
        if (rhs < lhs+tolerance){
            if (flag){
                j = P(j);
            }else{
                j--;
            }
            continue;
        }
        I = interval(lhs+tolerance, rhs);

        // The min value inside the square roots will be >= min(k^2)*tolerance,
        // so set inner_min to be less than that
        inner_min = k_sqr_min*tolerance/2;

        // Call the solver
        AllZeros(f_sigma, I, tolerance, zeros, unique, num_roots, error);

        // Display any error from the solver
        if (error){
            cout << endl << AllZerosErrMsg(error) << endl;
            return error;
        }

        // Check if there were any roots
        if (num_roots == 0){
            if (flag){
                j = P(j);
            }else{
                j--;
            }
            continue;
        }

        // Compute the equilibrium for each verified unique root
        for (u = 1; u <= num_roots; u++){
            if (unique[u]){
                // Calculate theta
                for (v = 0; v < n; v++){
                    theta = mid(asin(w[v]/(k[v]*sqrt(zeros[u]))));

                    // Shift to the correct quadrant so that the signs match
                    sgn = ((j >> (n-1-v)) & 1)*2 - 1;
                    if (cos(theta)*sgn < 0){
                        theta = pi - theta;
                    }
```

```
                    // Shift so the theta is in the interval (-pi, pi]
                    if (theta > pi){
                        theta -= 2*pi;
                    }else if (theta <= -1*pi){
                        theta += 2*pi;
                    }

                    // Display the equilibria
                    cout << theta;
                    if (v < n - 1){
                        cout << ", ";
                    }
                }
                cout << endl << endl;
            }else{
                // Include the possible roots that the solver couldn't verify
                // in the count of intervalus we cannot rule out
                num_given_up_intervals++;
            }
        }
        // Go to the next case
        j--;
    }

    // Display the number of problem intervals if there were any
    if (num_given_up_intervals > 0){
        cout << "The solver failed to determine if there was a root in ";
        cout << num_given_up_intervals << " intervals." << endl;
    }
    return 0;
}
```

Example A.1.1 *Running this code on a case with fifteen oscillators having two equilibria produces the following output in less than a second.*

```
Number of parameters: 15
w: 0.1 0.25 0.8 1 -1 -1 -1 1.25 -1.2 1.5 -2 -1.8 3 -4 4.1
k: 5 4.8 4.8 4.7 4.6 4 4 3.8 3.5 3.5 3 2.4 2.2 2 1.2
Tolerance: 1e-6

 5.804744025401485E-003,  1.511701176783033E-002,  4.839147968591000E-002,
 6.179156378059709E-002, -6.313662728886302E-002, -7.262271207465454E-002,
-7.262271207465454E-002,  9.561786293751634E-002, -9.967430024786283E-002,
 1.247096801268227E-001, -1.947185188847017E-001, -2.194334347573920E-001,
 4.069124532501765E-001, -6.193071711961917E-001,  1.441386302210293E+000
```

$$
\begin{array}{rrr}
5.834215320106978E{-}003, & 1.519376744623306E{-}002, & 4.863735895036847E{-}002, \\
6.210568543362181E{-}002, & -6.345760481327617E{-}002, & -7.299207557270629E{-}002, \\
-7.299207557270629E{-}002, & 9.610481566421319E{-}002, & -1.001820468718621E{-}001, \\
1.253461654739149E{-}001, & -1.957198998416935E{-}001, & -2.205658844969313E{-}001, \\
4.091015189688870E{-}001, & -6.229311023753289E{-}001, & 1.652393512117848E{+}000
\end{array}
$$

A.2 Rank Two Solver

The following is prototype Matlab code that finds the solutions to $\Psi_{K,\omega,+}$ when K is rank two. Moreover, if the input coupling matrix K is positive semidefinite, then the output will contain all the stable equilibria by Theorem 5.2.3. (A separate function would be needed to check each solution's stability properties.) The system of equations to determine $\Psi_{K,\omega,+}$ has been converted to equations over the real numbers using Proposition 5.1.9 and uses the bounds provided by Proposition 5.1.10. Note that the function call "Solve" in lines 27 and 36 is a stand-in for any algorithm that returns the roots of the system of equations.

```
1  function rank_two_all_pos_solver(K, omega)
2  % Solves the rank 2 KM in the all positive case
3  % K is expected to be a rank two symmetric matrix
4  % omega is expected to satisy IC1 & IC2 from Chapter 6
5
6      global L R w n;
7
8      % Initialize variables and tolerance constant
9      tol = 1e-6;
10     w = omega;
11     n = size(w,1);
12
13     % Use QR factorization to find L and R so that
14     % LR = K and L,R' are nx2
15     [L,R] = qr(K);
16     L = L(1:n,1:2);
17     R = R(1:2,1:n);
18
19     % Set up the bounds on psi for when q = 2
20     abs_row_sum = vecnorm(R,1,2);
21     bounds = [-abs_row_sum(1)  abs_row_sum(1);...  % x_1
22                            tol  abs_row_sum(2);...  % x_2 > 0
23               -abs_row_sum(1)  abs_row_sum(1);...  % y_1
24                             0                0];   % y_2 = 0
25
26     % Call a nonlinear system root finder for the q = 2 case
27     Psi_q2 = Solve(@rk2_real_eq, bounds);
28
29     % Set up the bounds on psi for when q = 1
```

```
    bounds = [tol abs_row_sum(1);... % x_1 > 0
                0                0;... % x_2 = 0
                0                0;... % y_1 = 0
                0                0];   % y_2 = 0

    % Call a nonlinear system root finder for the q = 1 case
    Psi_q1 = Solve(@rk2_real_eq, bounds);

    % Convert the solutions to Theta
    Theta = convert([Psi_q2 Psi_q1]);

    % Display the solutions
    disp(size(Theta,2))
    disp(Theta);


function z = rk2_real_eq(Psi)
% Implements the real version of the reformulation equations for
% the rank two case
% Psi is expected to be 4x1

    global L R w n;

    % Initialize z to the proper dimension
    z = Psi;

    % Initialize helper variables
    x = Psi(1:2); % real components of Psi
    y = Psi(3:4); % imag components of Psi

    r_hat_sqrs = (L*x).^2 + (L*y).^2;
    S = diag(n*w./r_hat_sqrs);
    T = diag(((r_hat_sqrs-n^2*w.^2).^(1/2))./r_hat_sqrs);

    % The equtions for which to find the roots
    z(1:2) = -x - R*(S*L*y + T*L*x);
    z(3:4) = -y + R*(S*L*x + T*L*y);


function Theta = convert(Psi)
% Converts psi -> theta

    global L w n;

    if isempty(Psi)
        Theta = [];
    else
        % Merge real and imag parts back together
```

```
        Psi = Psi(1:2,:) + Psi(3:4,:)*i;

        % Convert psi -> phi
        r = L*Psi;
        Phi = (n*w*i + (vecnorm(r,2,1).^2-n^2*w.^2).^(1/2))./conj(r);

        % Convert phi -> theta
        Theta = atan(imag(Phi)./real(Phi));
    end
```

APPENDIX

B

ARBITRARY RANK COUPLING WITH LOSS

It is interesting to note that the process of reformulation used in Chapter 5 can also be applied to the Kuramoto model with arbitrary rank coupling and loss matrices. However, the resulting system of equations and algorithm are equally if not more complicated than the other methods of solving, so it offers no computational advantages without further results.

The Kuramoto model with arbitrary rank coupling and loss matrices has the form

$$\frac{\mathrm{d}\theta_u}{\mathrm{d}t} = \omega_u - \frac{1}{n}\sum_{v=1}^{n} A_{uv}\cos(\theta_u - \theta_v) + B_{uv}\sin(\theta_u - \theta_v) \qquad \text{for } u = 1, 2, \ldots, n \tag{B.0.1}$$

where $n \geq 2$ is the number of oscillators, θ is the phase angles of the oscillators, ω is the natural frequencies of the oscillators, $A \in \mathbb{R}^{n\times n}$ is skew-symmetric and describes the loss, and $B \in \mathbb{R}^{n\times n}$ is symmetric and describes the coupling. As can be shown similarly to Section 1.1.1, setting $\frac{\mathrm{d}\theta_u}{\mathrm{d}t} = 0$ gives the equation for the equilibria when using the rotating reference frame. Thus the equations to find the equilibria can be simplified to

$$\omega_u = \frac{1}{n}\sum_{v=1}^{n} A_{uv}\cos(\theta_u - \theta_v) + B_{uv}\sin(\theta_u - \theta_v) \qquad \text{for } u = 1, 2, \ldots, n \tag{B.0.2}$$

We will take the following input conditions.

IC1: $\sum_{u=1}^{n} \omega_u = 0$

This is necessary to use the rotating reference frame.

IC2: $\omega_u \neq 0$ for $u = 1, 2, \ldots, n$

We will avoid cases where any oscillator has an intrinsic velocity equal to the average.

B.1 Finding

Note that if θ is a solution to Eq. 3.0.1, then $\theta + c$ is also a solution for any constant c. Thus, two solutions are called equivalent modulo shift if each component-wise difference is the same modulo 2π. In order to choose a representative, an output condition for $\theta \in (-\pi, \pi]^n$ is given below so that only one of these equivalent solutions is chosen. To set this up, a few variables will be defined. Let $C := A + Bi$. Then $C = LR$ for some $L, R^{\mathrm{T}} \in \mathbb{C}^{n\times\rho}$ where $\rho := \mathrm{rk}(C)$. Let $\phi_u := e^{i\theta_u}$ and let $\psi := R\phi$. Now we state the output condition as follows.

OC: $\quad \exists q \;\; \psi_q \in \mathbb{R}_+$ and $\psi_{q+1} = \cdots = \psi_\rho = 0$.

Remark B.1.1 ***OC*** *always produces a unique representative since if* $\psi = \vec{0}$*, then* $\gamma = C\phi = LR\phi = L\psi = \vec{0}$*, but* $\gamma_u \neq 0$ *as long as* ***IC2*** *is satisfied as shown in the proof of Lemma B.1.4.*

Let $C := A + Bi$ and $C = LR$ for some $L, R^{\mathrm{T}} \in \mathbb{C}^{n\times\rho}$ where $\rho := \mathrm{rk}(C)$. Let $\mathbb{U}$ be the unit circle in the complex plane, $\mathbb{U} := \{z \in \mathbb{C} : \; |z| = 1\}$. We can perform a series of transformations on $\Theta_{C,\omega}$ to transform the system.

Definition B.1.2 *Let* $\Theta_{C,\omega}$ *be the set of all solutions to Eq. B.0.2 satisfying* **OC***, that is,*

$$\Theta_{C,\omega} := \left\{ \theta \in (-\pi, \pi]^n : \underset{u}{\forall} \;\; \omega_u = \frac{1}{n}\sum_{v=1}^{n} A_{uv}\cos(\theta_u - \theta_v) + B_{uv}\sin(\theta_u - \theta_v) \wedge \mathbf{OC} \right\}$$

Lemma B.1.3 *Let*

$$f : \left([\theta_1, \ldots, \theta_n]^{\mathrm{T}}\right) \longmapsto \left[e^{i\theta_1}, \ldots, e^{i\theta_n}\right]^{\mathrm{T}}$$

Then we have

1. $f|_{\Theta_{C,\omega}}$ *is injective.*
2. $f\left(\Theta_{C,\omega}\right) = \Phi_{C,\omega}$ *where*

$$\Phi_{C,\omega} := \left\{ \phi \in \mathbb{U}^n : \underset{u}{\forall} \; n\omega_u = \mathrm{Re}\left(\phi_u \gamma_u^*\right) \wedge \mathbf{OC} \right\}$$

and where

$$\gamma := C\phi.$$

(Note that γ_u^* *denotes the complex conjugate of* γ_u*.)*

Proof.

1. Let $\theta, \theta' \in \Theta_{C,\omega}$ be such that $f(\theta) = f(\theta')$. Then for every u we have $e^{i\theta_u} = e^{i\theta'_u}$, and thus $\theta_u = \theta'_u$ since $\theta_u, \theta'_u \in (-\pi, \pi]$. Hence $\theta = \theta'$.

2. Let $\alpha := A\phi$ and $\beta := B\phi$. Note that z^* denotes the complex conjugate of z. Then

$$
\begin{aligned}
f\left(\Theta_{C,\omega}\right) &= \left\{f(\theta) \colon \theta \in \Theta_{C,\omega}\right\} \\
&= \left\{f(\theta) \colon \underset{u}{\forall}\ \omega_u = \frac{1}{n}\sum_{v=1}^{n} A_{uv}\cos(\theta_u - \theta_v) + B_{uv}\sin(\theta_u - \theta_v) \wedge \mathbf{OC}\right\} \\
&= \left\{f(\theta) \colon \underset{u}{\forall}\ n\omega_u = \mathrm{Re}\left(\sum_{v=1}^{n} A_{uv} e^{i(\theta_u - \theta_v)}\right) + \mathrm{Im}\left(\sum_{v=1}^{n} B_{uv} e^{i(\theta_u - \theta_v)}\right) \wedge \mathbf{OC}\right\} \\
&= \left\{\phi \in \mathbb{U}^n \colon \underset{u}{\forall}\ n\omega_u = \mathrm{Re}\left(\sum_{v=1}^{n} A_{uv}\phi_u\phi_v^*\right) + \mathrm{Im}\left(\sum_{v=1}^{n} B_{uv}\phi_u\phi_v^*\right) \wedge \mathbf{OC}\right\} \\
&= \left\{\phi \in \mathbb{U}^n \colon \underset{u}{\forall}\ n\omega_u = \mathrm{Re}\left(\phi_u\sum_{v=1}^{n} A_{uv}\phi_v^*\right) + \mathrm{Im}\left(\phi_u\sum_{v=1}^{n} B_{uv}\phi_v^*\right) \wedge \mathbf{OC}\right\} \\
&= \left\{\phi \in \mathbb{U}^n \colon \underset{u}{\forall}\ n\omega_u = \mathrm{Re}\left(\phi_u\alpha_u^*\right) + \mathrm{Im}\left(\phi_u\beta_u^*\right) \wedge \mathbf{OC}\right\} \\
&= \left\{\phi \in \mathbb{U}^n \colon \underset{u}{\forall}\ n\omega_u = \frac{1}{2}\left(\phi_u\alpha_u^* + \phi_u^*\alpha_u\right) + \frac{1}{2i}\left(\phi_u\beta_u^* - \phi_u^*\beta_u\right) \wedge \mathbf{OC}\right\} \\
&= \left\{\phi \in \mathbb{U}^n \colon \underset{u}{\forall}\ 2n\omega_u = \left(\phi_u\alpha_u^* + \phi_u^*\alpha_u\right) - \left(\phi_u\beta_u^* - \phi_u^*\beta_u\right)i \wedge \mathbf{OC}\right\} \\
&= \left\{\phi \in \mathbb{U}^n \colon \underset{u}{\forall}\ 2n\omega_u = \phi_u\left(\alpha_u^* - \beta_u^* i\right) + \phi_u^*\left(\alpha_u + \beta_u i\right) \wedge \mathbf{OC}\right\} \\
&= \left\{\phi \in \mathbb{U}^n \colon \underset{u}{\forall}\ n\omega_u = \frac{1}{2}\left(\phi_u\gamma_u^* + \phi_u^*\gamma_u\right) \wedge \mathbf{OC}\right\} \\
&= \left\{\phi \in \mathbb{U}^n \colon \underset{u}{\forall}\ n\omega_u = \mathrm{Re}(\phi_u\gamma_u^*) \wedge \mathbf{OC}\right\}
\end{aligned}
$$

where $\gamma := \alpha + \beta i = (A\phi + B\phi i) = (A + Bi)\phi = C\phi$.

■

Lemma B.1.4 *We have*

$$\Phi_{C,\omega} = \bigcup_{\sigma \in \{-,+\}^n} \Phi_{C,\omega,\sigma}$$

where

$$\Phi_{C,\omega,\sigma} := \left\{\phi \in \mathbb{U}^n \colon \underset{u}{\forall}\ \phi_u = \frac{n\omega_u + \sigma_u\sqrt{n^2\omega_u^2 - |\gamma_u|^2}}{\gamma_u^*} \wedge \mathbf{OC}\right\}$$

and $\gamma := C\phi$.

Proof. We have that

$$
\begin{aligned}
\Phi_{C,\omega} &= \left\{\phi \in \mathbb{U}^n \colon \underset{u}{\forall}\ n\omega_u = \mathrm{Re}(\phi_u\gamma_u^*) \wedge \mathbf{OC}\right\} \\
&= \left\{\phi \in \mathbb{U}^n \colon \underset{u}{\forall}\ 2n\omega_u = \phi_u\gamma_u^* + \phi_u^*\gamma_u \wedge \mathbf{OC}\right\}
\end{aligned}
$$

Note that $|\phi_u| = 1$, so $\phi_u \neq 0$. Thus multiplying through by ϕ_u gives us that

$$\begin{aligned}\Phi_{C,\omega} &= \left\{\phi \in \mathbb{U}^n : \underset{u}{\forall}\ 2n\omega_u\phi_u = \gamma_u^*\phi_u^2 + \gamma_u \wedge \mathbf{OC}\right\} \\ &= \left\{\phi \in \mathbb{U}^n : \underset{u}{\forall}\ 0 = \gamma_u^*\phi_u^2 - 2n\omega_u\phi_u + \gamma_u \wedge \mathbf{OC}\right\}\end{aligned}$$

For $\phi \in \Phi_C$, we have that $\gamma_u^* \neq 0$. Otherwise, we would have $0 = \gamma_u^*\phi_u^2 - 2n\omega_u\phi_u + \gamma_u = -2n\omega_u\phi_u$ and thus $\phi_u = 0$ or $\omega_u = 0$, a contradiction. Therefore we can use the quadratic formula to solve for ϕ_u.

$$\begin{aligned}\Phi_{C,\omega} &= \bigcup_{\sigma\in\{-,+\}^n} \left\{\phi \in \mathbb{U}^n : \underset{u}{\forall}\ \phi_u = \frac{2n\omega_u + \sigma_u\sqrt{4n^2\omega_u^2 - 4\gamma_u^*\gamma_u}}{2\gamma_u^*} \wedge \mathbf{OC}\right\} \\ &= \bigcup_{\sigma\in\{-,+\}^n} \left\{\phi \in \mathbb{U}^n : \underset{u}{\forall}\ \phi_u = \frac{n\omega_u + \sigma_u\sqrt{n^2\omega_u^2 - |\gamma_u|^2}}{\gamma_u^*} \wedge \mathbf{OC}\right\}\end{aligned}$$

■

Remark B.1.5 *Note that*

$$n\omega_u = \mathrm{Re}(\phi_u\gamma_u^*) = \mathrm{Re}\left(n\omega_u + \sigma_u\sqrt{n^2\omega_u^2 - |\gamma_u|^2}\right)$$

Therefore,

$$|\gamma_u|^2 \geq n^2\omega_u^2$$

Lemma B.1.6 *Let*

$$g : \phi \longmapsto R\phi.$$

Then we have that

1. $g|_{\Phi_{C,\omega,\sigma}}$ *is injective.*
2. $g(\Phi_{C,\omega,\sigma}) = \Psi_{C,\sigma}$ *where*

$$\Psi_{C,\omega,\sigma} := \left\{\psi \in \mathbb{C}^\rho : \underset{u}{\forall}\ \psi_u = \sum_{\nu=1}^{n} R_{u\nu}\frac{n\omega_u + \sigma_\nu\sqrt{n^2\omega_u^2 - \left|\hat{\gamma}_\nu\right|^2}}{\hat{\gamma}_\nu^*} \wedge \mathbf{OC}\right\}$$

and $\hat{\gamma} := L\psi$.

Proof.

1. Let $\phi, \phi' \in \Phi_{C,\sigma}$ be such that $g(\phi) = g(\phi')$. Then

$$\gamma = C\phi = LR\phi = Lg\left(\phi\right) = Lg(\phi') = LR\phi' = C\phi' = \gamma'$$

Thus

$$\phi_u = \frac{n\omega_u + \sigma_u\sqrt{n^2\omega_u^2 - |\gamma_u|^2}}{\gamma_u^*} = \frac{n\omega_u + \sigma_u\sqrt{n^2\omega_u^2 - |\gamma'_u|^2}}{(\gamma'_u)^*} = \phi'_u$$

2. Note that $g(\phi) = \psi$. Recall that $\gamma := C\phi$ and define $\hat{\gamma} := L\psi$. We have that

$$\begin{aligned}
g(\Phi_{C,\omega,\sigma}) &= \{g(\phi) : \phi \in \Phi_{C,\omega,\sigma}\} \\
&= \left\{ g(\phi) : \underset{u}{\forall}\, \phi_u = \frac{n\omega_u + \sigma_u\sqrt{n^2\omega_u^2 - |\gamma_u|^2}}{\gamma_u^*} \wedge \mathbf{OC} \right\} \\
&= \left\{ \psi \in \mathbb{C}^\rho : \psi = R\phi \wedge \underset{u}{\forall}\, \phi_u = \frac{n\omega_u + \sigma_u\sqrt{n^2\omega_u^2 - |\gamma_u|^2}}{\gamma_u^*} \wedge \mathbf{OC} \right\} \\
&= \left\{ \psi \in \mathbb{C}^\rho : \underset{u}{\forall}\, \psi_u = \sum_{v=1}^{n} R_{uv}\phi_v \wedge \underset{u}{\forall}\, \phi_u = \frac{n\omega_u + \sigma_u\sqrt{n^2\omega_u^2 - |\gamma_u|^2}}{\gamma_u^*} \wedge \mathbf{OC} \right\} \\
&= \left\{ \psi \in \mathbb{C}^\rho : \underset{u}{\forall}\, \psi_u = \sum_{v=1}^{n} R_{uv}\left(\frac{n\omega_v + \sigma_v\sqrt{n^2\omega_v^2 - |\gamma_v|^2}}{\gamma_v^*} \right) \wedge \right. \\
&\qquad \left. \underset{u}{\forall}\, \phi_u = \frac{n\omega_u + \sigma_u\sqrt{n^2\omega_u^2 - |\gamma_u|^2}}{\gamma_u^*} \wedge \mathbf{OC} \right\} \\
&= \left\{ \psi \in \mathbb{C}^\rho : \underset{u}{\forall}\, \psi_u = \sum_{v=1}^{n} R_{uv}\left(\frac{n\omega_v + \sigma_v\sqrt{n^2\omega_v^2 - |\hat{\gamma}_v|^2}}{\hat{\gamma}_v^*} \right) \wedge \mathbf{OC} \right\}
\end{aligned}$$

since

$$\gamma = C\phi = LR\phi = L\psi = \hat{\gamma}$$

■

Combining the above Lemmas gives the following reformulation of $\Theta_{C,\omega}$.

Theorem B.1.7 *We have*

$$\Theta_{C,\omega} = F\left(\bigcup_{\sigma \in \{-,+\}^n} G_\sigma\left(\Psi_{C,\omega,\sigma}\right) \right)$$

where

$$\begin{aligned}
F(\phi) &:= \left[\arg(\phi_1), \ldots, \arg(\phi_n)\right]^{\mathrm{T}} \\
G_\sigma(\psi) &:= \left[\frac{n\omega_u + \sigma_1\sqrt{n^2\omega_u^2 - |\hat{\gamma}_1|^2}}{\hat{\gamma}_1^*}, \ldots, \frac{n\omega_u + \sigma_n\sqrt{n^2\omega_u^2 - |\hat{\gamma}_n|^2}}{\hat{\gamma}_n^*} \right]^{\mathrm{T}}
\end{aligned}$$

and $\hat{\gamma} := L\psi$. Furthermore, F and G_σ are injective.

Proof. We split the proof into three parts.

1. $F = f^{-1}$ and is injective.

Recall that

$$f\left([\theta_1,\ldots,\theta_n]^{\mathrm{T}}\right)=\left[e^{i\theta_1},\ldots,e^{i\theta_n}\right]^{\mathrm{T}}.$$

Hence it is obvious that

$$f^{-1}\left(\phi\right)=\left[\arg\left(\phi_1\right),\ldots,\arg\left(\phi_n\right)\right]^{\mathrm{T}}=F\left(\phi\right)$$

and that F is injective.

2. $G_\sigma = g^{-1}|_{\Phi_{C,\omega,\sigma}}$ and is injective.

   Recall that

$$g\left(\phi\right)=R\phi.$$

   Let $g\left(\phi\right)=\psi$ where $\phi\in\Phi_{C,\omega,\sigma}$. Then

$$\gamma=C\phi=LR\phi=L\psi=\hat{\gamma},$$

   so

$$\phi_u=\frac{n\omega_u+\sigma_u\sqrt{n^2\omega_u^2-\left|\gamma_u\right|^2}}{\gamma_u^*}=\frac{n\omega_u+\sigma_u\sqrt{n^2\omega_u^2-\left|\hat{\gamma}_u\right|^2}}{\hat{\gamma}_u^*}=G_\sigma\left(\psi\right)_u.$$

   Furthermore, let $\psi,\psi'\in\Psi_{C,\sigma}$ be such that $G_\sigma\left(\psi\right)=G_\sigma\left(\psi'\right)$. Then

$$\hat{\gamma}=L\psi=LRG_\sigma\left(\psi\right)=LRG_\sigma\left(\psi'\right)=L\psi'=\hat{\gamma}'.$$

   Thus

$$\psi_u=\sum_{v=1}^{n}R_{uv}\frac{n\omega_u+\sigma_v\sqrt{n^2\omega_u^2-\left|\hat{\gamma}_v\right|^2}}{\hat{\gamma}_v^*}=\sum_{v=1}^{n}R_{uv}\frac{n\omega_u+\sigma_v\sqrt{n^2\omega_u^2-\left|\hat{\gamma}'_v\right|^2}}{\left(\hat{\gamma}'_v\right)^*}=\psi'_u.$$

3. $\Theta_C=F\left(\bigcup_{\sigma\in\{-1,+1\}^n}G_\sigma\left(\Psi_{C,\sigma}\right)\right)$.

   Combining Lemmas B.1.3, B.1.4, and B.1.6 with parts one and two above, we have

$$\Theta_{C,\omega}=f^{-1}\left(\Phi_{C,\omega}\right)=f^{-1}\left(\bigcup_{\sigma\in\{-,+\}^n}\Phi_{C,\omega,\sigma}\right)=f^{-1}\left(\bigcup_{\sigma\in\{-,+\}^n}g^{-1}\left(\Psi_{C,\omega,\sigma}\right)\right)$$

■

In order to make use of the reformulation theorem and turn it into an algorithm, we need to split up the equilibria in $\Psi_{C,\omega,\sigma}$ based on q in **OC**.

Definition B.1.8 *Let*

$$\Psi^q_{C,\omega,\sigma} := \left\{ \psi \in \mathbb{C}^r : \underset{u}{\forall} \left(\psi_u = \sum_{v=1}^{n} R_{uv} \frac{n\omega_u + \sigma_v \sqrt{n^2\omega_u^2 - \left|\hat{\gamma}_v\right|^2}}{\hat{\gamma}_v^*} \right) \wedge \psi_q > 0 \wedge \psi_{q+1} = \cdots = \psi_\rho = 0 \right\}$$

Note that $\Psi_{C,\omega,\sigma} = \bigcup_{q\in\{1,2,\ldots,\rho\}} \Psi^q_{C,\omega,\sigma}$.

We can convert $\Psi^q_{C,\omega,\sigma}$ to use equations over the real numbers and real variables by breaking the equations for ψ_u into real and imaginary components. This allows for solvers that function over only the real numbers to be used to find $\Psi_{K,\omega,\sigma}$. Moreover, we can bound the values of the real and imaginary components by the absolute row sums of R.

Proposition B.1.9 *Let* $x := \mathrm{Re}(\psi)$ *and* $y := \mathrm{Im}(\psi)$. *Then*

$$\begin{aligned}
\Psi^q_{C,\omega,\sigma} = \Big\{ x + y\,i : \; & x, y \in \mathbb{R}^n \wedge \\
& x = \mathrm{Re}(R)\big(S\,\mathrm{Re}(\hat{\gamma}) - T\,\mathrm{Im}(\hat{\gamma})\big) - \mathrm{Im}(R)\big(S\,\mathrm{Im}(\hat{\gamma}) + T\,\mathrm{Re}(\hat{\gamma})\big) \wedge \\
& y = \mathrm{Re}(R)\big(S\,\mathrm{Im}(\hat{\gamma}) + T\,\mathrm{Re}(\hat{\gamma})\big) + \mathrm{Im}(R)\big(S\,\mathrm{Re}(\hat{\gamma}) - T\,\mathrm{Im}(\hat{\gamma})\big) \wedge \\
& \underset{u>q}{\forall}\; x_u = y_u = 0 \wedge \\
& x_q > 0,\; y_q = 0 \Big\}
\end{aligned}$$

where

$$S := \begin{bmatrix} \frac{n\omega_1}{|\hat{\gamma}_1|^2} & & \\ & \ddots & \\ & & \frac{n\omega_1}{|\hat{\gamma}_1|^2} \end{bmatrix} \qquad T := \begin{bmatrix} \frac{\sigma_1\sqrt{|\hat{\gamma}_1|^2 - n^2\omega_1^2}}{|\hat{\gamma}_1|^2} & & \\ & \ddots & \\ & & \frac{\sigma_n\sqrt{|\hat{\gamma}_n|^2 - n^2\omega_n^2}}{|\hat{\gamma}_n|^2} \end{bmatrix}$$

$$\mathrm{Re}(\hat{\gamma}) = \mathrm{Re}(L)x - \mathrm{Im}(L)y \qquad \mathrm{Im}(\hat{\gamma}) = \mathrm{Im}(L)x + \mathrm{Re}(L)y$$

and

$$|\hat{\gamma}_u|^2 = \big(\mathrm{Re}(L)x - \mathrm{Im}(L)y\big)_u^2 + \big(\mathrm{Im}(L)x + \mathrm{Re}(L)y\big)_u^2$$

Proof. We have that

$$\begin{aligned}
x_u &= \mathrm{Re}(\psi_u) \\
&= \mathrm{Re}\left(\sum_{v=1}^{n} R_{uv} \frac{n\omega_v + \sigma_v\sqrt{n^2\omega_v^2 - |\hat{\gamma}_v|^2}}{\hat{\gamma}_v^*} \right) \\
&= \mathrm{Re}\left(\sum_{v=1}^{n} R_{uv}\hat{\gamma}_u \frac{n\omega_v + \sigma_v\sqrt{n^2\omega_v^2 - |\hat{\gamma}_v|^2}}{|\hat{\gamma}_v^2|} \right)
\end{aligned}$$

$$= \sum_{v=1}^{n} \frac{1}{|\hat{\gamma}_v^2|} \operatorname{Re}\left(n\omega_v R_{uv} \hat{\gamma}_v + \sigma_v R_{uv} \hat{\gamma}_v \sqrt{n^2\omega_v^2 - |\hat{\gamma}_v|^2} \right)$$

Recall that

$$|\hat{\gamma}_u|^2 \geq n^2 \omega_u^2$$

for any equilibria (see Remark B.1.5). Thus

$$\begin{aligned}
x_u &= \sum_{v=1}^{n} \frac{\operatorname{Re}(R_{uv})}{|\hat{\gamma}_v|^2} \operatorname{Re}\left(n\omega_v \hat{\gamma}_v + \sigma_v \hat{\gamma}_v i \sqrt{|\hat{\gamma}_v|^2 - n^2\omega_v^2} \right) \\
&\quad - \sum_{v=1}^{n} \frac{\operatorname{Im}(R_{uv})}{|\hat{\gamma}_v|^2} \operatorname{Im}\left(n\omega_v \hat{\gamma}_v + \sigma_v \hat{\gamma}_v i \sqrt{|\hat{\gamma}_v|^2 - n^2\omega_v^2} \right) \\
&= \sum_{v=1}^{n} \frac{\operatorname{Re}(R_{uv})}{|\hat{\gamma}_v|^2} \left(n\omega_v \operatorname{Re}(\hat{\gamma}_v) - \sigma_v \operatorname{Im}(\hat{\gamma}_v) \sqrt{|\hat{\gamma}_v|^2 - n^2\omega_v^2} \right) \\
&\quad - \sum_{v=1}^{n} \frac{\operatorname{Im}(R_{uv})}{|\hat{\gamma}_v|^2} \left(n\omega_v \operatorname{Im}(\hat{\gamma}_v) + \sigma_v \operatorname{Re}(\hat{\gamma}_v) \sqrt{|\hat{\gamma}_v|^2 - n^2\omega_v^2} \right)
\end{aligned}$$

Doing the same for the imaginary parts gives the following.

$$\begin{aligned}
y_u &= \operatorname{Im}(\phi_u) \\
&= \operatorname{Im}\left(\sum_{v=1}^{n} R_{uv} \frac{n\omega_v + \sigma_v \sqrt{n^2\omega_v^2 - |\hat{\gamma}_v|^2}}{\hat{\gamma}_v^*} \right) \\
&= \operatorname{Im}\left(\sum_{v=1}^{n} R_{uv} \hat{\gamma}_u \frac{n\omega_v + \sigma_v \sqrt{n^2\omega_v^2 - |\hat{\gamma}_v|^2}}{|\hat{\gamma}_v|^2} \right) \\
&= \sum_{v=1}^{n} \frac{1}{|\hat{\gamma}_v|^2} \operatorname{Im}\left(n\omega_v R_{uv} \hat{\gamma}_v + \sigma_v R_{uv} \hat{\gamma}_v \sqrt{n^2\omega_v^2 - |\hat{\gamma}_v|^2} \right) \\
&= \sum_{v=1}^{n} \frac{\operatorname{Re}(R_{uv})}{|\hat{\gamma}_v|^2} \operatorname{Im}\left(n\omega_v \hat{\gamma}_v + \sigma_v \hat{\gamma}_v i \sqrt{|\hat{\gamma}_v|^2 - n^2\omega_v^2} \right) \\
&\quad + \sum_{v=1}^{n} \frac{\operatorname{Im}(R_{uv})}{|\hat{\gamma}_v|^2} \operatorname{Re}\left(n\omega_v \hat{\gamma}_v + \sigma_v \hat{\gamma}_v i \sqrt{|\hat{\gamma}_v|^2 - n^2\omega_v^2} \right) \\
&= \sum_{v=1}^{n} \frac{\operatorname{Re}(R_{uv})}{|\hat{\gamma}_v|^2} \left(n\omega_v \operatorname{Im}(\hat{\gamma}_v) + \sigma_v \operatorname{Re}(\hat{\gamma}_v) \sqrt{|\hat{\gamma}_v|^2 - n^2\omega_v^2} \right) \\
&\quad + \sum_{v=1}^{n} \frac{\operatorname{Im}(R_{uv})}{|\hat{\gamma}_v|^2} \left(n\omega_v \operatorname{Re}(\hat{\gamma}_v) - \sigma_v \operatorname{Im}(\hat{\gamma}_v) \sqrt{|\hat{\gamma}_v|^2 - n^2\omega_v^2} \right)
\end{aligned}$$

Also, we have that

$$\hat{\gamma} = L\psi = L(x + yi) = (Lx) + (Ly)i$$

Therefore,

$$\begin{aligned}\mathrm{Re}(\hat{\gamma}) &= \mathrm{Re}((Lx)+(Ly)i) \\ &= \mathrm{Re}(L)x - \mathrm{Im}(L)y\end{aligned} \qquad \begin{aligned}\mathrm{Im}(\hat{\gamma}) &= \mathrm{Im}((Lx)+(Ly)i) \\ &= \mathrm{Im}(L)x + \mathrm{Re}(L)y\end{aligned}$$

and

$$|\hat{\gamma}_u|^2 = \left(\mathrm{Re}(L)x - \mathrm{Im}(L)y\right)_u^2 + \left(\mathrm{Im}(L)x + \mathrm{Re}(L)y\right)_u^2$$

Finally, we can combine the equations together using matrix multiplication giving the result. ■

Proposition B.1.10 *Let $x + y\,i \in \Psi^q_{K,\omega,\sigma}$ where $x, y \in \mathbb{R}^n$. Then $|x_u|,\ |y_u| \leq \sum_{v=1}^{n} |R_{uv}|$ for all u.*

Proof. Let $\psi \in \Psi_{K,\omega,\sigma}$ and let $x := \mathrm{Re}(\psi)$ and $y := \mathrm{Im}(\psi)$. Let $\phi := G_\sigma(\psi)$ and recall that $\phi \in \mathbb{U}^n$. Then we have that

$$\begin{aligned}|x_u|,\ |y_u| &\leq |\psi_u| \\ &= \left|\sum_{v=1}^{n} R_{uv}\phi_v\right| \\ &\leq \sum_{v=1}^{n} |R_{uv}\phi_v| \\ &= \sum_{v=1}^{n} |R_{uv}||\phi_v| \\ &\leq \sum_{v=1}^{n} |R_{uv}|\end{aligned}$$

■

Putting all of this together gives an algorithm to find $\Theta_{C,\omega}$. The following algorithm relies on three functions. **rk**(M) returns the rank of the matrix M. **Factor**(M) returns two matrices $L, R^{\mathrm{T}} \in \mathbb{C}^{n\times\rho}$ where ρ is the rank of M such that $LR = M$. **Solve**$(\Psi^q_{C,\omega,\sigma}, B)$ returns all the solutions satisfying the set of equations in $\Psi^q_{C,\omega,\sigma}$ (i.e., the elements of that set) within the bounding box B.

Algorithm B.1.11

In: $A, B \in \mathbb{R}^{n\times n}$ *and* $\omega \in \mathbb{R}^n$ *satisfying* **IC1–IC2**.

Out: $\Theta_{C,\omega}$

1. $C \leftarrow A + Bi$
2. $\rho \leftarrow$ **rk**(C)
3. $L, R \leftarrow$ **Factor**(C)

4. $\Theta_{C,\omega} \leftarrow \{\}$

5. *For* $q \in \{1,\dots,\rho\}$

   (a) $B \leftarrow$ *bounds for* x *and* y *using Proposition B.1.10*

   (b) *For* $\sigma \in \{-1,+1\}^n$

      i. $S \leftarrow \textbf{Solve}\left(\Psi^q_{C,\omega,\sigma}\right)$

      ii. $\Theta_{C,\omega} \leftarrow \Theta_{C,\omega} \cup F\left(G_\sigma(S)\right)$

6. *Return* $\Theta_{C,\omega}$

B.2 Stability

In this section we will show that the Jacobian for Eq. B.0.2 can be rewritten in terms of the variables used for the reformulation. Also, note that J is still symmetric in this case, so all its eigenvalues are real. To determine the stability of an equilibrium in the orbitally stable sense (see Section 1.1.2), one could to show that J evaluated at an equilibrium has $n-1$ negative eigenvalues. Likewise, if J has a positive eigenvalue, then that equilibrium cannot be stable [9].

Proposition B.2.1 *The Jacobian of Eq. B.0.1 in terms of the variables used in the reformulation in the previous section is given by*

$$J = \begin{bmatrix} \sigma_1\sqrt{|\gamma_1|^2 - n^2\omega_1^2} & & \\ & \ddots & \\ & & \sigma_n\sqrt{|\gamma_n|^2 - n^2\omega_n^2} \end{bmatrix} - \frac{1}{n}\Big[A_{uv}\operatorname{Im}\left(\phi_u\phi_v^*\right) - B_{uv}\operatorname{Re}\left(\phi_u\phi_v^*\right)\Big]_{(u,v)}$$

Proof. The Jacobian for Eq. B.0.2 can be directly computed as

$$J = -\frac{1}{n}\begin{bmatrix} \sum_{v=1}^{n} -A_{1v}\sin(\theta_1-\theta_v) + B_{1v}\cos(\theta_1-\theta_v) & & \\ & \ddots & \\ & & \sum_{v=1}^{n} -A_{nv}\sin(\theta_n-\theta_v) + B_{nv}\cos(\theta_n-\theta_v) \end{bmatrix}$$

$$-\frac{1}{n}\Big[A_{uv}\sin(\theta_u-\theta_v) - B_{uv}\cos(\theta_u-\theta_v)\Big]_{(u,v)}$$

Recall that $\phi := e^{i\theta}$, $\alpha := A\phi$, $\beta := B\phi$, $C := A + Bi$, and $\gamma := C\phi$. Following the same procedure as Lemma B.1.3, we have that

$$\sum_{v=1}^{n} -A_{uv}\sin(\theta_u-\theta_v) + B_{uv}\cos(\theta_u-\theta_v) = \sum_{v=1}^{n} -A_{uv}\operatorname{Im}\left(e^{i(\theta_u-\theta_v)}\right) + B_{uv}\operatorname{Re}\left(e^{i(\theta_u-\theta_v)}\right)$$

$$
\begin{aligned}
&= \operatorname{Im}\left(-\phi_u \sum_{v=1}^{n} A_{uv}\phi_v^*\right) + \operatorname{Re}\left(\phi_u \sum_{v=1}^{n} B_{uv}\phi_v^*\right) \\
&= -\operatorname{Im}\left(\phi_u \alpha_u^*\right) + \operatorname{Re}\left(\phi_u \beta_u^*\right) \\
&= -\frac{1}{2i}\left(\phi_u \alpha_u^* - \phi_u^* \alpha_u\right) + \frac{1}{2}\left(\phi_u \beta_u^* + \phi_u^* \beta_u\right) \\
&= \frac{1}{2i}\left(\phi_u\left(-\alpha_u^* + \beta_u^* i\right) + \phi_u^*\left(\alpha_u + \beta_u i\right)\right) \\
&= \frac{1}{2i}\left(-\phi_u \gamma_u^* + \phi_u^* \gamma_u\right) \\
&= \operatorname{Im}\left(\phi_u^* \gamma_u\right) \\
&= -\operatorname{Im}\left(\phi_u \gamma_u^*\right)
\end{aligned}
$$

Therefore we can rewrite the Jacobian as

$$
J = \frac{1}{n}\begin{bmatrix} \operatorname{Im}\left(\phi_1 \gamma_1^*\right) & & \\ & \ddots & \\ & & \operatorname{Im}\left(\phi_n \gamma_n^*\right) \end{bmatrix} - \frac{1}{n}\left[A_{uv}\operatorname{Im}\left(\phi_u \phi_v^*\right) - B_{uv}\operatorname{Re}\left(\phi_u \phi_v^*\right)\right]_{(u,v)}
$$

Finally, from $\Phi_{C,\omega,\sigma}$, we see that $\operatorname{Im}\left(\phi_u \gamma_u^*\right) = \sigma_u \sqrt{|\gamma_u|^2 - n^2\omega_u^2}$. ■