

STARS

University of Central Florida
STARS

Electronic Theses and Dissertations, 2004-2019

2005

Phase Synchronization In Three-dimensional Lattices And Globally Coupled Populations Of Nonidentical Rossler Oscillators

Limin Qi

University of Central Florida



Part of the [Mathematics Commons](#)

Find similar works at: <https://stars.library.ucf.edu/etd>

University of Central Florida Libraries <http://library.ucf.edu>

This Masters Thesis (Open Access) is brought to you for free and open access by STARS. It has been accepted for inclusion in Electronic Theses and Dissertations, 2004-2019 by an authorized administrator of STARS. For more information, please contact STARS@ucf.edu.

STARS Citation

Qi, Limin, "Phase Synchronization In Three-dimensional Lattices And Globally Coupled Populations Of Nonidentical Rossler Oscillators" (2005). *Electronic Theses and Dissertations, 2004-2019*. 605.

<https://stars.library.ucf.edu/etd/605>



Showcase of Text, Archives, Research & Scholarship

PHASE SYNCHRONIZATION IN THREE-DIMENSIONAL LATTICES AND
GLOBALLY COUPLED POPULATIONS OF NONIDENTICAL ROSSLER
OSCILLATORS

by

Limin Qi

B.S. Xian Institute of Science and Technology, 1982

A thesis submitted in partial fulfillment of the requirements
for the degree of Master of Science
in the Department of Mathematics
in the College of Arts and Sciences
at the University of Central Florida
Orlando, Florida

Fall Term
2005

© 2005 Limin Qi

ABSTRACT

A study on phase synchronization in large populations of nonlinear dynamical systems is presented in this thesis. Using the well-known Rossler system as a prototypical model, phase synchronization in one oscillator with periodic external forcing and in two-coupled nonidentical oscillators was explored at first. The study was further extended to consider three-dimensional lattices and globally coupled populations of nonidentical oscillators, in which the mathematical formulation that represents phase synchronization in the generalized N -coupled Rossler system was derived and several computer programs that perform numerical simulations were developed. The results show the effects of coupling dimension, coupling strength, population size, and system parameter on phase synchronization of the various Rossler systems, which may be applicable to studying phase synchronization in other nonlinear dynamical systems as well.

ACKNOWLEDGMENTS

I would like to express my gratitude to Dr. C. Schober for her valuable guidance and encouragement. She introduced me to the field of nonlinear dynamics, which has been a very active and attractive research field during the last few decades.

I thank very much my husband and daughter for their patience. I would have not completed this thesis without their constant support and great love.

TABLE OF CONTENTS

LIST OF FIGURES.....	vii
CHAPTER ONE INTRODUCTION	1
CHAPTER TWO NOTION AND COMPUTATION OF PHASE SYNCHRONIZATION	4
2.1 Rossler System: A Prototypic Model of Nonlinear Dynamics	4
2.2 Mathematical Formulation.....	10
2.3 Process for Numerical Simulations	16
CHAPTER THREE PHASE SYNCHRONIZATION IN SIMPLE ROSSLER OSCILLATORS.....	17
3.1 One Oscillator with Periodic External Forcing.....	17
3.2 Two-Coupled Nonidentical Oscillators.....	21
CHAPTER FOUR PHASE SYNCHRONIZATION IN COMPLEX ROSSLER OSCILLATORS.....	28
4.1 Lattices of Nonidentical Oscillators	28
4.2 Populations of Globally Coupled Nonidentical Oscillators.....	32
4.3 Effect of Topological Parameter	35
CHAPTER FIVE CONCLUSION	38
APPENDIX A COMPUTER PROGRAMS FOR NUMERICAL SIMULATIONS.....	40
A.1 Trajectory of Rossler System on Phase Plane.....	41
A.2 One Oscillator with Periodic External Forcing and Fixed Topological Parameter	42

A.3 Two-Coupled Nonidentical Oscillators with Fixed Topological Parameter	43
A.4 One Oscillator with Periodic External Forces and Variation of Topological Parameter	45
A.5 Two-Coupled Nonidentical Oscillators with Variation of Topological Parameter	47
A.6 Lattices of Nonidentical Oscillators with Fixed Topological Parameter	49
A.7 Populations of Globally Coupled Nonidentical Oscillators with Fixed Topological Parameter	52
A.8 Lattices of Nonidentical Oscillators with Variation of Topological Parameter.....	55
A.9 Populations of Globally Coupled Nonidentical Oscillators with Variation of Topological Parameter	58
REFERENCES.....	63

LIST OF FIGURES

Fig. 2.1(a). Trajectory from Rossler system projected on the x - y plane ($a = 0.09$), which forms a limit cycle attractor and performs periodic motion.....	7
Fig. 2.1(b). Trajectory from Rossler system projected on the x - y plane ($a = 0.10$), which forms a limit cycle attractor and performs periodic motion.....	7
Fig. 2.1(c). Trajectory from Rossler system projected on the x - y plane ($a = 0.15$), which forms a phase coherent attractor and performs smeared cycle motion.....	8
Fig. 2.1(d). Trajectory from Rossler system projected on the x - y plane ($a = 0.19$), which forms a phase coherent attractor and performs smeared cycle motion.....	8
Fig. 2.1(e). Trajectory from Rossler system projected on the x - y plane ($a = 0.20$), which forms a phase coherent attractor and performs smeared cycle motion.....	9
Fig. 2.1(f). Trajectory from Rossler system projected on the x - y plane ($a = 0.25$), which forms a phase incoherent attractor and performs funnel motion.....	9
Fig. 2.2(a). X component of Rossler system vs. time ($a = 0.15$).	11
Fig. 2.2(b). Y component of Rossler system vs. time ($a = 0.15$).....	12
Fig. 3.1(a). Phase synchronization region for one Rossler oscillator with the external force applied to the x -coordinate.	19
Fig. 3.1(b). Phase synchronization region for one Rossler oscillator with the external force applied to the y -coordinate.	19
Fig. 3.1(c). Phase synchronization region for one Rossler oscillator with the external forces applied to the x - and y -coordinates.....	20

Fig. 3.1(d). Phase synchronization region for one Rossler oscillator with the external forces applied to the x -, y -, and z -coordinates.	20
Fig. 3.2(a). Phase synchronization region for two-coupled Rossler oscillators along the x -coordinate.....	23
Fig. 3.2(b). Phase synchronization region for two-coupled Rossler oscillators along the y -coordinate.....	23
Fig. 3.2(c). Phase synchronization region for two-coupled Rossler oscillators along the x - and y -coordinates.....	24
Fig. 3.2(d). Phase synchronization region for two-coupled Rossler oscillators along the x -, y -, and z -coordinates.	24
Fig. 3.3. Phase synchronization region for one Rossler oscillator with the external force applied to the y -coordinate and variation of a ($E = 0.5$).	27
Fig 3.4. Phase synchronization region for two-coupled Rossler oscillators along the x -, y -, and z -coordinates with variation of a ($\Delta\omega = 0.015$).	27
Fig. 4.1. Standard deviation of frequencies in one-, two-, and three-dimensional lattices of 500 nonidentical Rossler oscillators ($a = 0.15$).	31
Fig. 4.2. Standard deviation of frequencies in one-dimensional(x -coordinate) lattices of nonidentical Rossler oscillators with various population sizes ($a = 0.15$). ...	31
Fig. 4.3. Standard deviation of frequencies in one-, two-, and three-dimensional populations of 500 globally coupled nonidentical Rossler oscillators ($a = 0.15$).	34

Fig. 4.4. Standard deviation of frequencies in one-dimensional (x -coordinate)	
populations of globally coupled nonidentical Rossler oscillators with various	
population sizes ($a = 0.15$).	35
Fig. 4.5. Standard deviation of frequencies in a three-dimensional lattice of 200	
nonidentical Rossler oscillators with variation of a	37
Fig. 4.6. Standard deviation of frequencies in a three-dimensional population of 100	
globally coupled nonidentical Rossler oscillators with variation of a	37

CHAPTER ONE

INTRODUCTION

Synchronization of coupled oscillators is one of the major topics in nonlinear dynamics and has been well studied for the classical periodic systems (e.g., the Van de Pol and Duffing oscillators) as well as for the modern chaotic systems (e.g., the Lorenz and Rossler [1] oscillators).

Different types of synchronization may appear in chaotic systems. One possibility is due to the interaction of at least two identical chaotic systems their states (i.e., amplitudes and phases) can coincide. This type of synchronization is referred to as complete (full) synchronization. However, if the parameters of the coupled oscillators slightly mismatch, their states cannot coincide exactly. Instead, they are only closed to each other. Such synchronization is denoted as generalized synchronization. Phase synchronization of chaotic oscillators occurs when there is locking between the phases of two coupled systems while the amplitudes vary chaotically in time [2].

In this thesis we study phase synchronization in large populations of nonlinear dynamical systems using the Rossler system. To do so the Rossler system is generalized from a single uncoupled system to the N -coupled system with different types of coupling, i.e., we generalize from one oscillator with external forcing, to two-coupled oscillators, to lattices of oscillators, and finally to populations of globally coupled oscillators. We are interested in finding the phase synchronization regimes, where all oscillators of the systems are phase locked. We also investigate the effects of the coupling dimension,

coupling strength, population size, and topological parameter of the Rossler system on phase synchronization.

With widespread studies of nonlinear dynamics over last several decades, the rigorous mathematical methods combined physical ideas on synchronization were developed. The Rossler system is a prototypical model of nonlinear dynamics that has been broadly used in studying phase synchronization of chaotic oscillators. There have been many research papers on this topic in the current literature. Most of these studies concentrated on only one oscillator with external forcing (see Refs. [12, 16]) or two coupled oscillators (see Refs. [2-6, 13, 15]). Few of the papers addressed the issue of large populations of coupled oscillators (see Refs. [7-11, 14]). Further, they primarily considered the effects of coupling strength between the oscillators and did not consider the effects of coupling dimension or population size. Typically, only identical oscillators were examined rather than nonidentical oscillators (see Refs. [9-11]). None of these studies considered the effect of the system parameter.

We generalize the study of phase synchronization in Rossler system as follows:

1. For one oscillator with a periodic external force in [12] and two coupled oscillators in [3], we extend them from one-dimensional to two- and three-dimensional coupling.
2. For lattices of oscillators in [7] and populations of globally coupled oscillators in [8], we extend them from one-dimensional to two- and three-dimensional coupling. Meanwhile, we consider the effect of population size on phase synchronization.

3. For the above four coupled Rossler systems, we perform numerical simulations by considering the effect of the topological parameter of the Rossler system on phase synchronization (i.e., we consider $a = 0.05 \sim 0.20$ rather than only $a = 0.15$ in the existing publications).

The rest of this thesis is organized as follows. In Chapter 2 we briefly introduce chaotic systems using the Rossler system. We present the ideas of phase and frequency locking in chaotic systems and the general procedure for the numerical simulations. Chapter 3 is devoted to determining the phase synchronization regimes for the case of one oscillator with periodic external forcing and the case of two-coupled when the coupling dimension and the topological parameter of the Rossler system are varied. Chapter 4 extends the approaches presented in Chapter 3 to lattices and populations of globally coupled nonidentical oscillators, where the effects of coupling dimension, coupling strength, population size, and topological parameter are studied. Chapter 5 concludes the current study and indicates the future direction of the research. In Appendix A, several computer programs are attached. These programs were coded in Microsoft Visual C++ and were executed on a PC. They are easily modified to perform more numerical simulations on phase synchronization and to study other nonlinear dynamical systems for those who are familiar with the basic aspect of the numerical ODE and C/C++ programming language,

CHAPTER TWO

NOTION AND COMPUTATION OF PHASE SYNCHRONIZATION

2.1 Rossler System: A Prototypic Model of Nonlinear Dynamics

From many experimental observations and numerical simulations in the studies of nonlinear dynamics, it has been discovered that some rather simple dynamical systems can generate very complex chaotic motion. The term “chaotic motion” means that the precise behavior of the system can not be determined for very long time in contrast to periodic or quasi-periodic motion which can be determined. The phase spaces of chaotic systems do not form simple geometric portraits like the limit cycles formed by periodic or quasi-periodic systems. The complex geometric portraits formed by chaotic motion are called strange attractors in contrast to the limit cycles that are called simple attractors.

Both regular and chaotic systems can be modeled using ordinary differential equations. In representing regular systems we have always considered systems that can be represented on the phase plane, i.e., with two independent variables. This is the minimal dimension of the phase space for a limit cycle oscillatory, but it is not enough for chaotic systems. Because the trajectories cannot intersect in phase space, it is not possible to encounter anything more complex than a limit cycle on a phase plane. Therefore, an autonomous chaotic system must have at least three dimensions, i.e., the state of such oscillators must be described by at least three independent variables.

The German physicist O. Rossler in 1976 [1] proposed such a system, which has exactly three independent variables and is a prototypical example of autonomous chaotic oscillators.

$$\begin{aligned}
\dot{x} &= -y - z \\
\dot{y} &= x + ay \\
\dot{z} &= b + z(x - c)
\end{aligned} \tag{2.1}$$

Here a , b , and c are system parameters. Furthermore, we consider the Rossler system with the natural frequency ω as follows

$$\begin{aligned}
\dot{x} &= -\omega y - z \\
\dot{y} &= \omega x + ay \\
\dot{z} &= b + z(x - c)
\end{aligned} \tag{2.2}$$

which can be rewritten in the form

$$\begin{aligned}
\ddot{y} - a\dot{y} + \omega^2 y &= -\omega z \\
\dot{z} + cz &= b + z(\dot{y} - ay) / \omega
\end{aligned} \tag{2.3}$$

where the parameter a , which represents a negative damping coefficient, determines the topology of the Rossler system. Thus, we call it a topological parameter [7]. In this study we fix $b = 0.4$ and $c = 8.5$, and change the value of a . The computer program in Section A.1 was coded to simulate trajectories of the Rossler system on the x - y plane. The different trajectories on the x - y plane obtained by changing the value of a are illustrated in Figs. 2.1(a)-(f). When a is less than or equal to 0.1, the trajectories form limit cycle attractors (see Figs. 2.1(a) and (b)). The Rossler oscillator performs periodic motion. When a is between 0.1 and 0.2, the trajectories become phase-coherent attractors (see Figs. 2.1(c)-(e)). The Rossler oscillator performs smeared cycle motion. When a is larger 0.2, the trajectories develop funnels (see Fig. 2.1(f)). The Rossler oscillator develops phase-incoherent attractors. For the limit cycle and phase-coherent attractors the phase of the Rossler oscillator is well defined, i.e., all approaches to defining the phase, which will

be outlined in Section 2.2, give similar results. We use the coincidence of the observed frequencies of the Rossler oscillators as the criterion of phase synchronization of coupling or forcing Rossler systems. This will be introduced in Section 2.2. We select 0.15 as the value of a or restrict a from 0.05 to 0.2 as we study the effect of a on phase synchronization of the Rossler systems demonstrated in Chapters 3 and 4. On the other hand, the phase for the phase-incoherent Rossler attractor is ill-defined and special approaches are needed to characterize the phase synchronization indirectly. So far appropriate methods for determining phase synchronization of the phase-incoherent Rossler attractor have not been found in the existing literature.

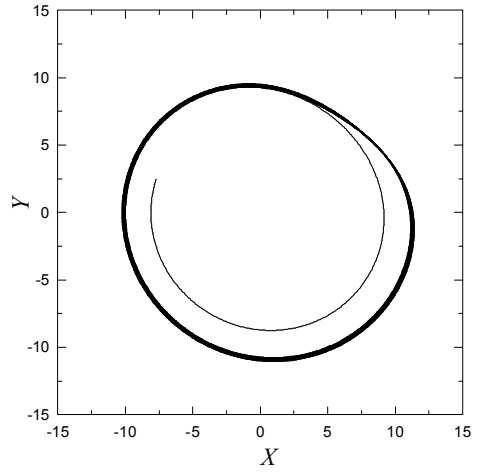


Fig. 2.1(a). Trajectory from Rossler system projected on the x - y plane ($a = 0.09$), which forms a limit cycle attractor and performs periodic motion.

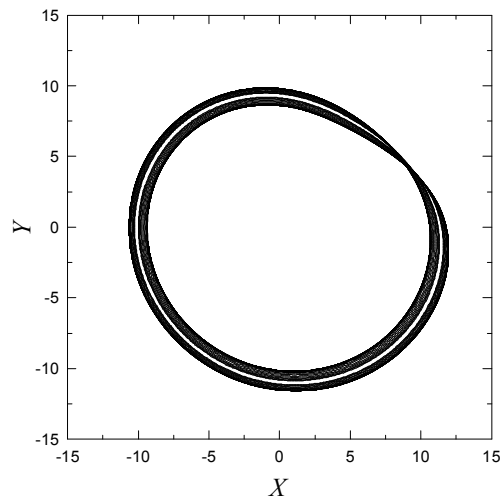


Fig. 2.1(b). Trajectory from Rossler system projected on the x - y plane ($a = 0.10$), which forms a limit cycle attractor and performs periodic motion.

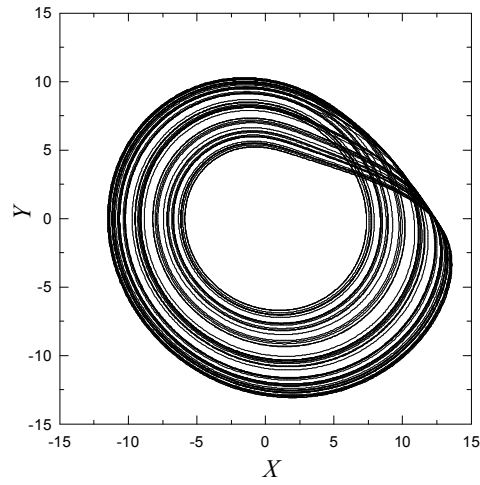


Fig. 2.1(c). Trajectory from Rossler system projected on the x - y plane ($a = 0.15$), which forms a phase coherent attractor and performs smeared cycle motion.

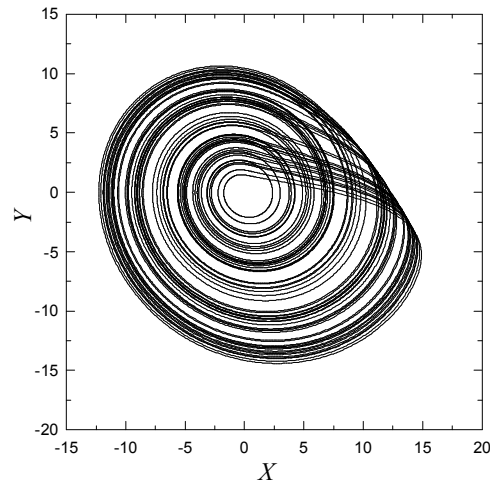


Fig. 2.1(d). Trajectory from Rossler system projected on the x - y plane ($a = 0.19$), which forms a phase coherent attractor and performs smeared cycle motion.

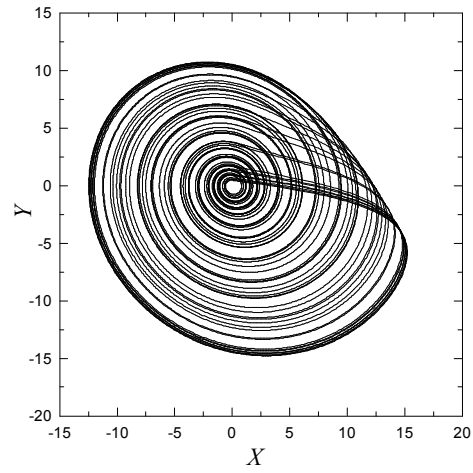


Fig. 2.1(e). Trajectory from Rossler system projected on the x - y plane ($a = 0.20$), which forms a phase coherent attractor and performs smeared cycle motion.

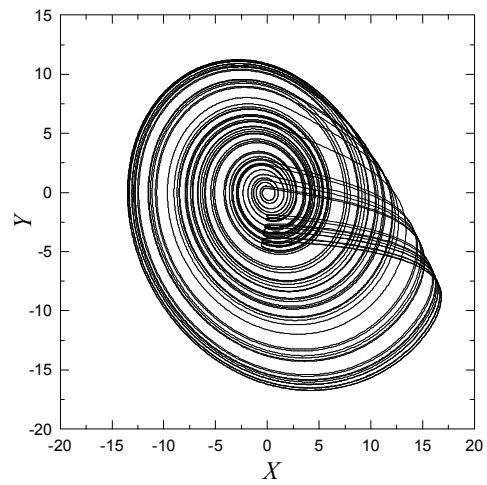


Fig. 2.1(f). Trajectory from Rossler system projected on the x - y plane ($a = 0.25$), which forms a phase incoherent attractor and performs funnel motion.

2.2 Mathematical Formulation

Most studies in nonlinear dynamics have represented phase synchronization with two approaches: one is based on the Poincare map and the other is based on a phase space projection [7]. Both of them can be used to compute the observed frequencies of oscillators.

In the first approach we find a projection of the attractor on the phase plane (x , y) such that the plot looks like a smeared limit cycle, i.e., trajectory rotates around the origin. This means that we can define a phase, attributing to each rotation the 2π phase increase:

$$\phi_M = 2\pi \frac{t - t_n}{t_{n+1} - t_n} + 2\pi n, \quad t_n \leq t \leq t_{n+1} \quad (2.4)$$

where t_n is the time of the n th crossing of the secant surface. Note that this definition yields the correct phase in the cases of either periodic or phase coherent chaotic oscillators.

If a nonlinear dynamic system is considered, we can use the coincidence of the observed frequencies as the criterion of phase synchronization. We emphasize that the mean observed frequency, Ω , of a system can be calculated as it follows from Eq. (2.4)

$$\Omega = \lim_{t \rightarrow \infty} 2\pi \frac{N_t}{t} \quad (2.5)$$

where N_t is the number of crossing the Poincare section during observation time t . This method can be straightforwardly applied to the observed time history. In the

simplest case we can take for N_t the number of local maxima or minima of $x(t)$ or $y(t)$. This is because the Rossler system exhibits the phase coherent features in both the x - and y -ordinates in the certain range of its topological parameter a ($a \leq 0.2$) as shown in Figs. 2.2(a) and (b). From where the numbers of either local maxima or minima of $x(t)$ or $y(t)$ are counted as 16 and the observation time is 100. Thus, $N_t = 16$ as $t = 100$. Then we can calculate Ω by using Eq. (2.5).

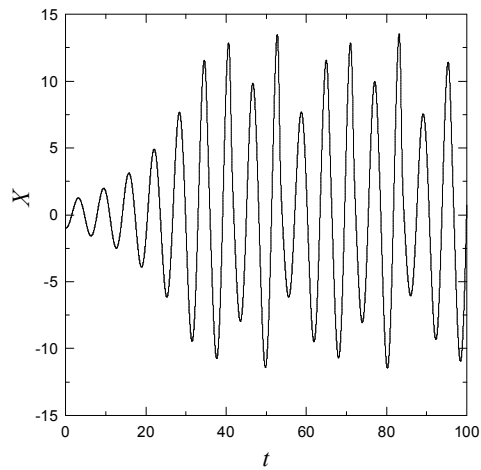


Fig. 2.2(a). X component of Rossler system vs. time ($a = 0.15$).

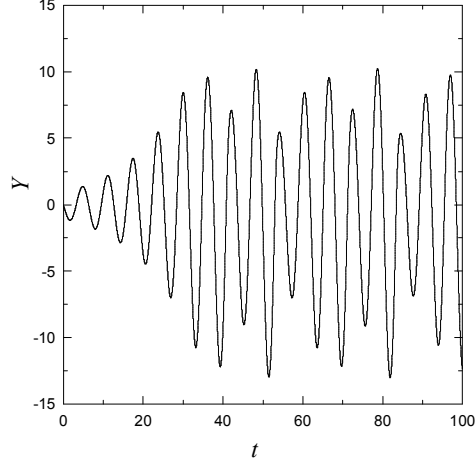


Fig. 2.2(b). Y component of Rossler system vs. time ($a = 0.15$).

If two or more coupled nonlinear dynamic systems are considered, the above expression can be generalized as

$$\Omega_i = \lim_{t \rightarrow \infty} 2\pi \frac{N_t^i}{t} \quad (2.6)$$

where the index $i = 1, \dots, N$ denotes the i th oscillator of a system having N -oscillators. N_t^i is the number of crossing the Poincare section during observation time t for the i th oscillator.

For a system of two-coupled oscillators we simply calculate the difference of two mean observed frequencies, $\Delta\Omega = |\Omega_1 - \Omega_2|$, as demonstrated in Section 3.2, where $\Delta\Omega \cong 0$ implies the occurrence of phase synchronization. On the other hand,

for a system of N -coupled oscillators we determine the standard deviation of mean frequencies σ as

$$\sigma = \sqrt{\frac{\sum_{i=1}^N (\Omega_i - \bar{\Omega})^2}{N-1}} \quad (2.7)$$

where the total mean of mean observed frequency for all N oscillators $\bar{\Omega}$ is found by

$$\bar{\Omega} = \frac{1}{N} \sum_{i=1}^N \Omega_i \quad (2.8)$$

Note that σ is not only dependent on Ω_i , but also the coupling strength, population size, coupling dimension, and system parameter (i.e., topological parameter of the Rossler system). We will demonstrate the effects of these parameters in Chapter 4, in which phase synchronization occurs as $\sigma \cong 0$.

In the second approach we introduce the phase as the angle between the projection of the phase point on the plane and a given direction on the plane as

$$\phi_i = \tan^{-1} \left(\frac{y_i}{x_i} \right) \quad (2.9)$$

Then the mean observed frequency defined as the average of $d\phi_i/dt$ over a large period of time coincides with a straightforward definition of the observed frequency via the average number of crossings of a Poincare surface per unit time. Therefore, the observed frequency based on phase angle ϕ_i during observation time T can be defined as

$$\Omega_i = \lim_{T \rightarrow \infty} \frac{\phi_i(T) - \phi_i(0)}{T} \quad (2.10)$$

where the index $i=1, \dots, N$ denotes the i th oscillator of a system having N -oscillators. Then we can obtain the standard deviation of mean observed frequencies by using Eqs (2.7) and (2.8).

Our experience with the numerical experiments shows that the second approach (Eq. (2.10)) is more time consuming than the first one (Eqs. (2.5) and (2.6)). This is because many trigonometric functions are involved in computation during the execution of programs.

In this study we generalize the Rossler system from a single uncoupled system to the N -coupled system ($3N$ ordinary differential equations) as follows

$$\begin{aligned}\dot{x}_i &= -\omega_i y_i - z_i + c_x \varepsilon X_i \\ \dot{y}_i &= \omega_i x_i + a y_i + c_y \varepsilon Y_i \\ \dot{z}_i &= b + z_i(x_i - c) + c_z \varepsilon Z_i\end{aligned}\tag{2.11}$$

where index $i=1, \dots, N$ denotes the i th oscillator having the natural frequency ω_i and three coupling components X_i , Y_i , and Z_i . ε is the coupling strength. c_x , c_y , and c_z stand for coupling dimension coefficient for the x -, y -, and z -coordinate, respectively. If $c_x=1$ and $c_y=c_z=0$, there are only one-dimensional coupling along the x -coordinate, and so on; If $c_x=c_y=1$ and $c_z=0$, there are two-dimensional couplings along the x - and y -coordinates, and so on; Likewise, if $c_x=c_y=c_z=1$, there are three-dimensional couplings along the x -, y -, and z -coordinates.

For one oscillator by periodic external forces, $i=1$,

$$\varepsilon X_1 = \varepsilon Y_1 = \varepsilon Z_1 = E \cos \nu t\tag{2.12}$$

where E and ν denote the amplitude and frequency of external forces, respectively.

For two-coupled oscillators, $i = 1, 2$,

$$X_1 = x_2 - x_1, \quad Y_1 = y_2 - y_1, \quad Z_1 = z_2 - z_1, \quad (2.13)$$

$$X_2 = x_1 - x_2, \quad Y_2 = y_1 - y_2, \quad Z_2 = z_1 - z_2.$$

For lattices of oscillators, $i=1, \dots, N$, we consider the effects of nearest neighbor oscillators,

$$X_i = x_{i+1} - 2x_i + x_{i-1}$$

$$Y_i = y_{i+1} - 2y_i + y_{i-1} \quad (2.14)$$

$$Z_i = z_{i+1} - 2z_i + z_{i-1}$$

For populations of globally coupled oscillators, $i=1, \dots, N$, we consider the effects of all of the oscillators

$$X_i = \frac{1}{N-1} \sum_{j=1}^N (x_j - x_i)$$

$$Y_i = \frac{1}{N-1} \sum_{j=1}^N (y_j - y_i) \quad (2.15)$$

$$Z_i = \frac{1}{N-1} \sum_{j=1}^N (z_j - z_i)$$

Note that both lattices and globally coupled populations of oscillators can be reduced to two-coupled oscillators when $i = 2$. These different types of coupling have been added to the computer programs attached in Appendix A. The initial conditions for all the four systems are set as $x_i(0) = -1$, $y_i(0) = 0$, and $z_i(0) = 0$, where $i=1, \dots, N$. The boundary conditions are not necessary except in lattices. The boundary conditions for the system of lattices will be given in Section 4.1 The results for the numerical simulations will be shown in Chapters 3 and 4.

2.3 Process for Numerical Simulations

Based on the mathematical formulation represented in Section 2.2, nine computer programs for numerical simulations in this study have been developed in Microsoft Visual C++ and executed on a PC. They are attached in Appendix A. Each of these program includes two subprograms and the main program. In the first subprogram the Rossler oscillator itself or a system of one oscillator with periodic external forcing, two-coupled oscillators, lattice of oscillators, and population of globally coupled oscillators is input. In the second subprogram the Forth-Order Runge-Kutta method is used to solve the systems of ordinary differential equations. In the main program a random number generator produces the uniformly random natural frequencies for the Rossler system. Then the main program calls the second subprogram to solve the systems of the differential equations. In the process of solving the Rossler systems, the main program computes the maxima of $x(t)$ or $y(t)$ for N_t^i until iterating N steps. Finally, the main program computes the total mean frequency and standard deviation for each oscillator.

CHAPTER THREE

PHASE SYNCHRONIZATION IN SIMPLE ROSSLER OSCILLATORS

After introducing the notion and computation of phase synchronization in nonlinear dynamic systems in Chapter 2, we will study two simple systems of the Rossler oscillators in this chapter. The first system is a system of one oscillator with periodic external forcing while the second one is a system of two-coupled nonidentical oscillators. We consider the effects of the coupling dimension, coupling strength, and topological parameter of the Rossler system. Also, we consider the effects of the amplitude and frequency of external forces for the first system and the effect of the frequency mismatch for the second system on phase synchronization in the Rossler system.

3.1 One Oscillator with Periodic External Forcing

For one Rossler oscillator with periodic external forcing, we substitute Eq. (2.12) into Eq. (2.11) with $\omega_1 = 1$ as follows

$$\begin{aligned}\dot{x} &= -y - z + c_x E \cos \nu t \\ \dot{y} &= x + ay + c_y E \cos \nu t \\ \dot{z} &= b + z(x - c) + c_z E \cos \nu t\end{aligned}\tag{3.1}$$

where E and ν stand for the amplitude and frequency of the periodic external force, respectively. The value of the system parameters a , b , and c were given in Section 2.1.

c_x , c_y , and c_z are coupling dimension coefficients as described in Section 2.2. The computer program in Section A.2 was coded to compute the mean observed frequency of the oscillator using Eq. (2.5) and the difference between the frequency of the external force and mean observed frequency, $\nu - \Omega$.

Figures 3.1(a)-(d) show that $\nu - \Omega$ depends on the amplitude E and frequency of the external force ν . Phase synchronization regions correspond to the plateaus, $\nu - \Omega \cong 0$. From these figures we find that the oscillator can not be coupled by the external forces when $E = 0$. Thus, there not exist phase synchronization region for $E = 0$. As the value of E increases, the phase synchronization region becomes large gradually until E reaches its critical values, $E = 0.8$, where $\nu - \Omega$ blows up. It indicates that the effect of E is equivalent to the coupling strength ε for two-coupled oscillators as described later.

Next, we observe that the range of ν for the phase synchronization regions is between 0.95 and 1.05 approximately. Then we rewrite ν as $\nu = \nu_0 \pm \Delta\nu$, i.e., $\nu = 1 \pm 0.05$. It shows that the effect of $\Delta\nu$ is similar to that of the frequency mismatch $\Delta\omega$ for two-coupled oscillators.

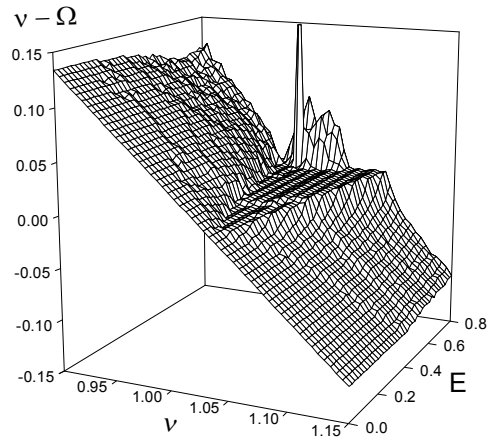


Fig. 3.1(a). Phase synchronization region for one Rossler oscillator with the external force applied to the x -coordinate.

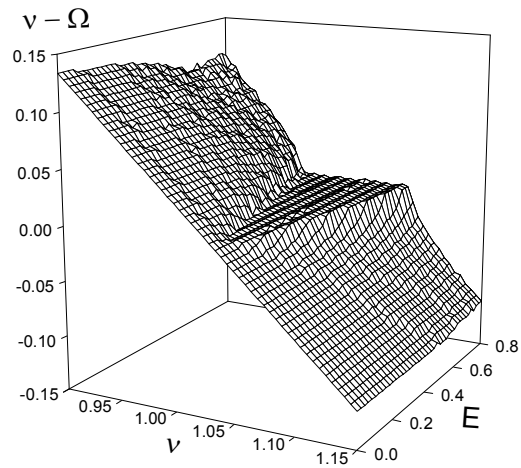


Fig. 3.1(b). Phase synchronization region for one Rossler oscillator with the external force applied to the y -coordinate.

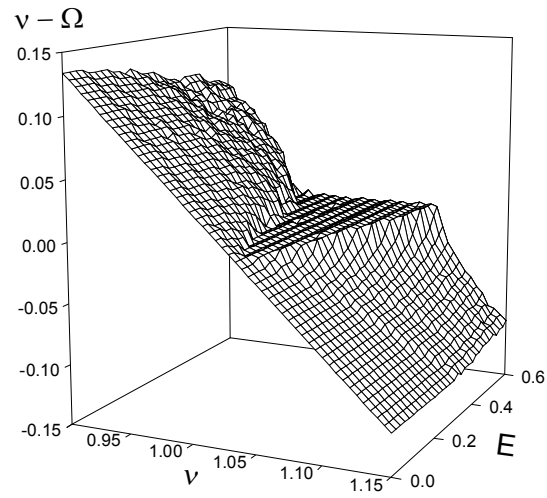


Fig. 3.1(c). Phase synchronization region for one Rossler oscillator with the external forces applied to the x - and y -coordinates.

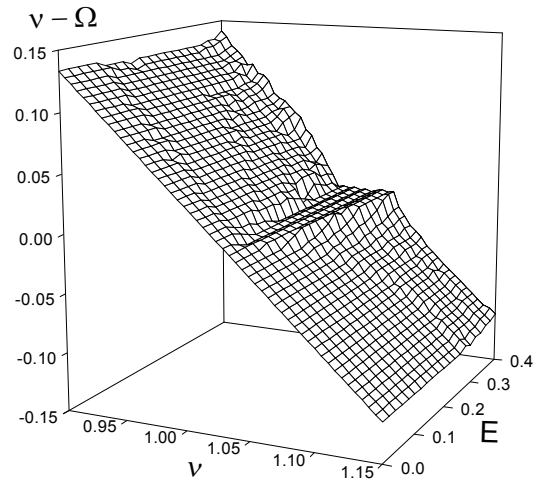


Fig. 3.1(d) Phase synchronization region for one Rossler oscillator with the external forces applied to the x -, y -, and z -coordinates.

Finally, we compare the phase synchronization region for one oscillator with external forces along different coordinates. From Figs. 3.1(a)-(d) we can see that the oscillator with one-dimensional external force forms the largest phase synchronization region with $E = 0.8$ (see Figs. 3.1(a) and (b)). The oscillator with two-dimensional external forcing forms the medium phase synchronization region with $E = 0.6$ (see Fig. 3.1(c)) while the oscillator with three-dimensional external forcing forms the smallest phase synchronization region with $E = 0.4$ (see Fig. 3.1(d)). It shows that phase synchronization of one oscillator with external forces becomes more difficult to attain as the dimension increases.

3.2 Two-Coupled Nonidentical Oscillators

For two-coupled nonidentical Rossler oscillators, we substitute Eq. (2.13) into Eq. (2.11) as follows

$$\begin{aligned}
\dot{x}_1 &= -\omega_1 y_1 - z_1 + c_x \varepsilon(x_2 - x_1) \\
\dot{y}_1 &= \omega_1 x_1 + ay_1 + c_y \varepsilon(y_2 - y_1) \\
\dot{z}_1 &= b + z_1(x_1 - c) + c_z \varepsilon(z_2 - z_1) \\
\dot{x}_2 &= -\omega_2 y_2 - z_2 + c_x \varepsilon(x_1 - x_2) \\
\dot{y}_2 &= \omega_2 x_2 + ay_2 + c_y \varepsilon(y_1 - y_2) \\
\dot{z}_2 &= b + z_2(x_2 - c) + c_z \varepsilon(z_1 - z_2)
\end{aligned} \tag{3.2}$$

where the subscripts 1 and 2 denote the oscillator 1 and 2, respectively. The value of the system parameters a , b , and c were given in Section 2.1. ε is the coupling

strength. c_x , c_y , and c_z are coupling dimension coefficients as described in Section 2.2. Since the two-coupled Rossler oscillators are nonidentical, we introduce two different natural frequencies for them, i.e., $\omega_1 = \omega_0 + \Delta\omega$ and $\omega_2 = \omega_0 - \Delta\omega$. Here $\omega_0 = 1$ is the normalized natural frequency and $\Delta\omega$ is the natural frequency mismatch between the two-coupled oscillators. The computer program in Section A.3 was coded to compute the mean observed frequency of each of two oscillators using Eq. (2.6) and the difference of two mean observed frequencies of nonidentical Rossler oscillators, $\Delta\Omega = |\Omega_1 - \Omega_2|$.

Figures 3.2(a)-(d) show that $\Delta\Omega$ depends on the coupling strength ε and the frequency mismatch $\Delta\omega$. Phase synchronization regions here correspond to the plateaus, $\Delta\Omega \cong 0$. From these figures we see that both oscillators are not coupled when $\varepsilon = 0$ and $\Delta\omega > 0$. Thus, for these values of ε and $\Delta\omega$, there does not exist a phase synchronization region.

On the other hand, phase synchronization regions become largest when $\Delta\omega = 0$, which forms a system of two-coupled identical oscillators. From Figs. 3.2(a) and (b) we see that for $\varepsilon = 0$ and $\Delta\omega \neq 0$, $\Delta\Omega \neq 0$. For $\varepsilon = 0.05$ and $0 \leq \Delta\omega \leq 0.01$, as well as $\varepsilon = 0.15$ and $0 \leq \Delta\omega \leq 0.01$, $\Delta\Omega = 0$. As the coupling strength ε increases, the values of $\Delta\omega$ that yield $\Delta\Omega = 0$ increases, and phase synchronization is more likely to occur for larger values of $\Delta\omega$. This indicates that stronger phase synchronization occurs when there is stronger interaction.

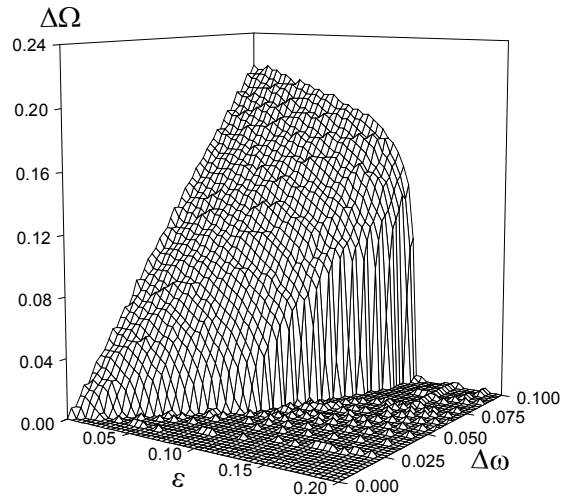


Fig. 3.2(a). Phase synchronization region for two-coupled Rossler oscillators along the x-coordinate.

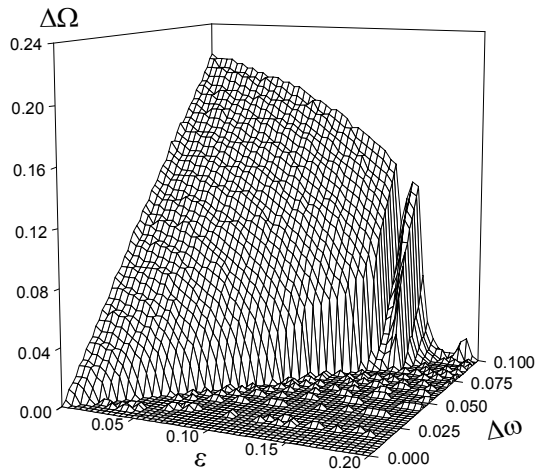


Fig. 3.2(b). Phase synchronization region for two-coupled Rossler oscillators along the y-coordinate.

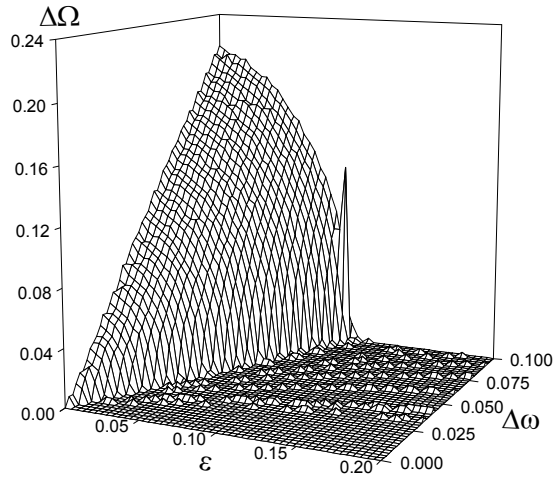


Fig. 3.2(c). Phase synchronization region for two-coupled Rossler oscillators along the x - and y -coordinates.

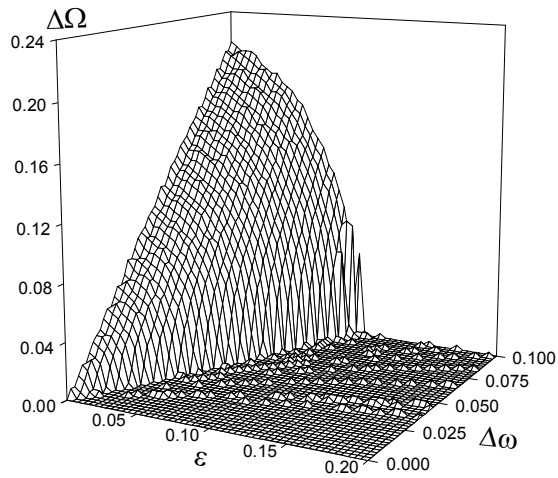


Fig. 3.2(d). Phase synchronization region for two-coupled Rossler oscillators along the x -, y -, and z -coordinates.

Finally, we compare the phase synchronization regions for two-coupled oscillators in different dimensions. From Figs. 3.2(a)-(d) we see that two-coupled oscillators along the x - or y -coordinate forms the smallest phase synchronization regions (see Figs. 3.2(a) and (b)). The system of two-coupled oscillators along the x - and y -coordinates forms the medium region (see Fig. 3.2(c)) while the system of two-coupled oscillators along the x -, y -, and z -coordinates forms the largest region (see Fig. 3.2(d)). This shows as the coupling dimension increases the region in the ε and $\Delta\omega$ for phase synchronization of two-coupled nonidentical oscillators increases.

3.3 Effect of Topological Parameter

In the previous two sections we have shown the effects of the amplitude and frequency of periodic external forces on a system of one oscillator and the effects of the coupling strength and frequency mismatch on a system of two-coupled nonidentical oscillators. Also we have shown the effect of the coupling dimension on both systems. Now we consider the effect of the topological parameter a on both systems by varying the value of a from 0.05 to 0.2. The computer program in Section A.4 was coded to simulate this effect in one oscillator by external forces while the computer program in Section A.5 was coded to simulate this effect in two-coupled nonidentical oscillators.

Figure 3.3 illustrates the phase synchronization region obtained for one oscillator when the external force is applied to the y -coordinate with the variation of a , where we fix the value of E as 0.5. Figure 3.4 illustrates the synchronization region

for two-coupled nonidentical oscillators along the x -, y -, and z -coordinates with the variation of a , where we fix the value of $\Delta\omega$ as 0.015. From Figs. 3.3 and 3.4, we see that phase synchronization, determined when $\nu - \Omega \cong 0$, has only a small dependence on the topological parameter a . Regardless of the value of a in Fig. 3.3, $\nu - \Omega \neq 0$ when $\nu < 0.98$ and $\nu > 1.05$. So phase synchronization of one oscillator with external forcing depends more on variation in the frequency of external force ν than in the topological parameter a . In Fig. 3.4, $\Delta\Omega \neq 0$ when $\varepsilon < 0.025$ and $\Delta\Omega = 0$ when $\varepsilon \geq 0.025$, which forms a rectangular strip. Therefore, phase synchronization of two-coupled oscillators depends on both the coupling strength ε and the topological parameter a .

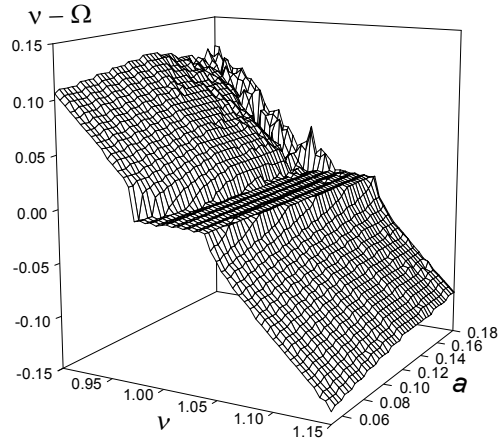


Fig. 3.3. Phase synchronization region for one Rossler oscillator with the external force applied to the y -coordinate and variation of a ($E = 0.5$).

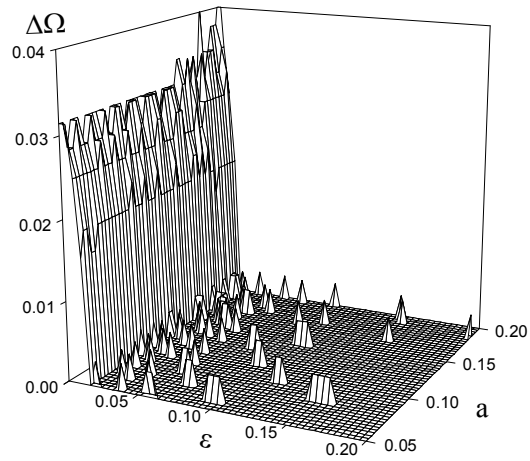


Fig 3.4. Phase synchronization region for two-coupled Rossler oscillators along the x -, y -, and z -coordinates with variation of a ($\Delta\omega = 0.015$).

CHAPTER FOUR

PHASE SYNCHRONIZATION IN COMPLEX ROSSLER OSCILLATORS

In this chapter we extend the Rossler system from the cases of one oscillator with external forcing and two-coupled oscillators to lattices and populations of globally coupled oscillators. Meanwhile, we consider these as nonidentical natural frequencies of the oscillators that are randomly and uniformly distributed. As we mentioned before, three-dimensional lattices and N -coupled populations of nonidentical Rossler oscillators have not been investigated yet. Additionally, there are very few publications on one- or two-dimensional lattices or coupled populations of oscillators in the current literature. Typically, identical oscillators were studied and the value of the topological parameter a for the Rossler system was fixed at $a = 0.15$. In this chapter, we will study phase synchronization in three-dimensional lattices and globally coupled populations of nonidentical Rossler oscillators for various values of the coupling dimension, population size, and system parameter instead of considering the coupling strength only.

4.1 Lattices of Nonidentical Oscillators

A lattice of oscillators is a natural generalization of the system of two-coupled oscillators as described in Section 2.2. For lattices of nonidentical Rossler oscillators, we substitute Eq. (2.14) into Eq. (2.11) as follows

$$\begin{aligned}
\dot{x}_i &= -\omega_i y_i - z_i + c_x \varepsilon (x_{i+1} - 2x_i + x_{i-1}) \\
\dot{y}_i &= \omega_i x_i + ay_i + c_y \varepsilon (y_{i+1} - 2y_i + y_{i-1}) \\
\dot{z}_i &= b + z_i (x_i - c) + c_z \varepsilon (z_{i+1} - 2z_i + z_{i-1})
\end{aligned} \tag{4.1}$$

with the free boundary conditions

$$\begin{aligned}
x_0 &= x_1, \quad x_{N+1} = x_N \\
y_0 &= y_1, \quad y_{N+1} = y_N \\
z_0 &= z_1, \quad z_{N+1} = z_N
\end{aligned} \tag{4.2}$$

where the index $i=1, \dots, N$ denotes the i th oscillator in a lattice. ω_i is the natural frequency of the i th oscillator, which are taken as uniformly distributed random numbers in the range 0.98 ± 0.04 . The values of the system parameters a , b , and c were given in Section 2.1. The coefficients c_x , c_y , and c_z are the coupling dimension coefficients as described in Section 2.2. The computer program in Section A.6 computes the mean observed frequency for each of the N oscillators using Eq. (2.6), the total mean of mean observed frequency of all the N oscillators using Eq. (2.8), and the standard deviation of mean observed frequencies of all the oscillators, $\sigma(\varepsilon)$, using Eq. (2.7). Phase synchronization occurs when $\sigma(\varepsilon) = 0$.

Due to the randomness of the natural frequencies of the oscillators in lattices and populations of globally coupled oscillators, we have performed a large number of numerical simulations. To determine the effect of the different coupling dimensions with the same natural frequencies on phase synchronization, we first generated a set of randomly natural frequencies, which was used to compute the standard deviation of mean observed frequencies of one-(x- or y-coordinate), two-(x- and y-coordinates),

and three-dimensional (all three coordinates) lattices, respectively. Figure 4.1 shows the standard deviation of frequencies $\sigma(\varepsilon)$ in one-, two- and three-dimensional lattices of 500 nonidentical Rossler oscillators with $a = 0.15$. From Fig. 4.1, we see that the graphs of $\sigma(\varepsilon)$ for two- and three-dimensional lattices coincide. It indicates that the z -coupling dimension does not impact phase synchronization for the Rossler system. On the other hand, we see that the critical values of the coupling strength ε , which makes $\sigma(\varepsilon) = 0$, for one-dimensional lattices (either x - or y -coordinate) are larger than that for two- and three-dimensional lattices. It shows that two- and three-dimensional lattices are easier to be synchronized than one-dimensional lattices.

Next, we compute the standard deviation of mean observed frequencies in one-dimensional (x -coordinate) lattices with population size $N = 100, 300, 700$, and 1000 and $a = 0.15$. Figure 4.2 shows the numerical results graphically. We see from Fig. 4.2 that the critical value of the coupling strength varies with the population size, i.e., $\varepsilon \cong 0.1$ as $N = 100$, $\varepsilon \cong 0.21$ as $N = 300$, $\varepsilon \cong 0.46$ as $N = 700$, and $\varepsilon \cong 0.62$ as $N = 1000$, respectively. It shows that the critical value of ε increases as the population size (i.e., the number of the oscillators in a lattice) increases. However, we have to point out that the critical values of the coupling strength are not constants with these population sizes. They are distributed in a certain range because we generate different natural frequencies randomly every time. Mostly, they are distributed in the range of $0.1 \sim 0.6$.

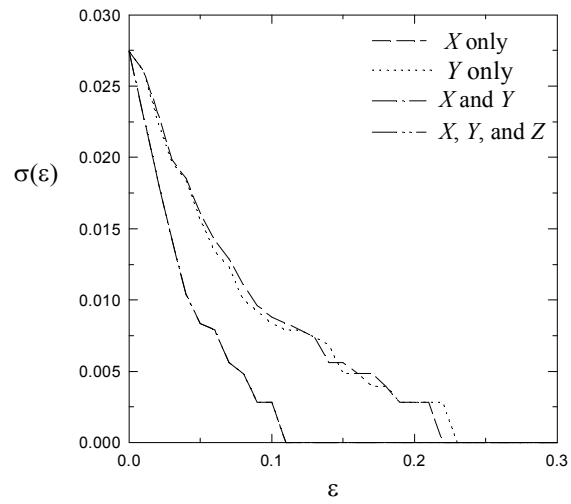


Fig. 4.1. Standard deviation of frequencies in one-, two-, and three-dimensional lattices of 500 nonidentical Rossler oscillators ($a = 0.15$).

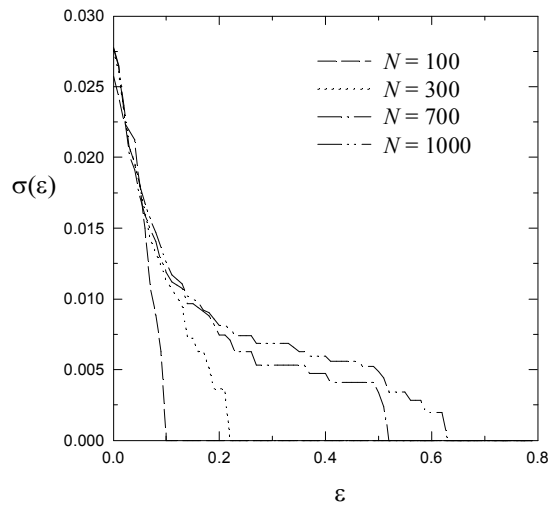


Fig. 4.2. Standard deviation of frequencies in one-dimensional(x -coordinate) lattices of nonidentical Rossler oscillators with various population sizes ($a = 0.15$).

4.2 Populations of Globally Coupled Nonidentical Oscillators

Often oscillators not only form a regular lattice with nearest-neighbor coupling and but also interact with many other oscillators. As a result, the effect of mutual synchronization of two-coupled oscillators described in Section 3.2, can be further generalized to the system with N -coupled oscillators instead of lattices of oscillators as described in Section 4.1. For populations of globally coupled nonidentical Rossler oscillators we substitute Eq. (2.15) into Eq. (2.11) as follows

$$\begin{aligned}
 \dot{x}_i &= -\omega_i y_i - z_i + \frac{c_x \varepsilon}{N-1} \sum_{j=1}^N (x_j - x_i) \\
 \dot{y}_i &= \omega_i x_i + a y_i + \frac{c_y \varepsilon}{N-1} \sum_{j=1}^N (y_j - y_i) \\
 \dot{z}_i &= b + z_i (x_i - c) + \frac{c_z \varepsilon}{N-1} \sum_{j=1}^N (z_j - z_i)
 \end{aligned} \tag{4.3}$$

where the index $i, j = 1, \dots, N$ denote the i th and j th oscillators in a population, respectively, and ω_i is the natural frequency of the i th oscillator, which are taken as uniformly distributed random numbers in the range 0.98 ± 0.04 . The value of the system parameters a , b , and c were given in Section 2.1. The coefficients c_x , c_y , and c_z are coupling dimension coefficients as described in Section 2.2. The computer program in Section A.7 computes the mean observed frequency for each of the N oscillators using Eq. (2.6), the total mean of the mean observed frequency of all the N oscillators using Eq. (2.8), and the standard deviation of the mean observed frequencies of all the oscillators, $\sigma(\varepsilon)$, using Eq. (2.7). Phase synchronization occurs when $\sigma(\varepsilon) = 0$.

As we mentioned in Section 4.1, we have performed a large number of numerical simulations due to the randomness of the natural frequencies of the oscillators. To determine the effect of the different coupling dimensions with the same natural frequencies on phase synchronization, we first generated a set of randomly natural frequencies, which was used to compute the standard deviation of mean observed frequencies of one- (x - or y -coordinate), two- (x - and y -coordinates), and three-dimensional (all three coordinates) populations of 500 globally coupled nonidentical Rossler oscillators with $a = 0.15$. Figure 4.3 shows the numerical results graphically. From Fig. 4.3, we see that the graphs of $\sigma(\varepsilon)$ for two- and three-dimensional populations coincide. It indicates again that the z -coupling dimension does not impact phase synchronization for the Rossler system. On the other hand, we see that the critical values of the coupling strength ε , which makes $\sigma(\varepsilon) = 0$, for one-dimensional populations globally coupled oscillators (either x - or y -coordinate) are larger than that for two- and three-dimensional populations globally coupled oscillators. It shows that two- and three-dimensional populations of globally coupled oscillators are easier to be synchronized than one-dimensional populations of globally coupled oscillators.

Next, we compute the standard deviation of mean observed frequencies in one-dimensional (x -coordinate) lattices with population size $N = 100, 300, 700$, and 1000 and $a = 0.15$. Figure 4.4 shows the numerical results graphically. We see from Fig. 4.4 that the critical value of the coupling strength is approximately equal to 0.025 even though we set N as 100, 300, 700, and 1000. It shows that the critical value of ε

is independent of the population size N . However, we have to point out that the critical values of the coupling strength are not constants with these population sizes. They are distributed in a certain range because we generate different natural frequencies randomly every time. Mostly, they are distributed in the range of 0.021 ~ 0.027.

Finally, we find by comparing Fig. 4.1 to Fig. 4.3 and by comparing Fig. 4.2. to Fig. 4.4 that the range of the critical value of ε in lattices is 0.1 ~ 0.62 while the range of the critical value of ε in globally coupled populations is around 0.011 ~ 0.027. This indicates that the populations of globally coupled nonidentical oscillators are easier to generate phase synchronization than the lattices. In other words, the lattices of oscillators need stronger coupling strength to be synchronized than the populations of globally coupled oscillators.

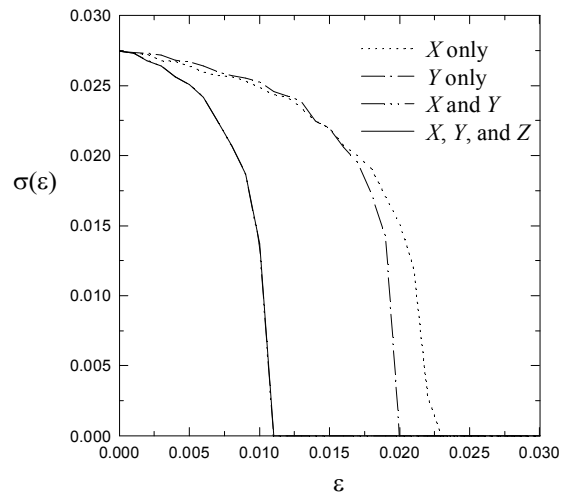


Fig. 4.3. Standard deviation of frequencies in one-, two-, and three-dimensional populations of 500 globally coupled nonidentical Rossler oscillators ($a = 0.15$).

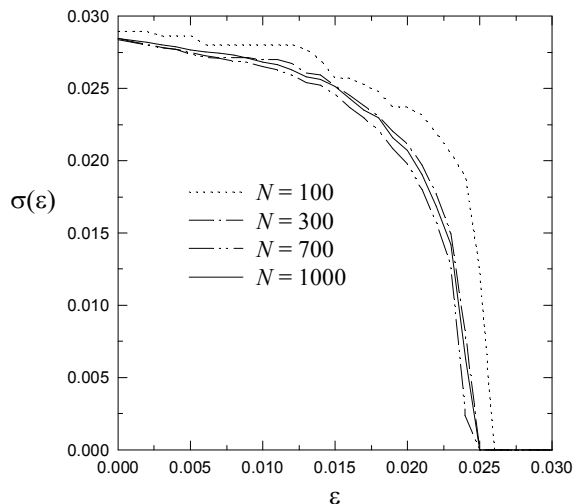


Fig. 4.4. Standard deviation of frequencies in one-dimensional (x -coordinate) populations of globally coupled nonidentical Rossler oscillators with various population sizes ($a = 0.15$).

4.3 Effect of Topological Parameter

In the previous two sections we have shown the effects of the coupling dimension, coupling strength, and population size to the systems of both lattices and globally coupled populations of nonidentical oscillators for a fixed $a = 0.15$. Now we consider the effect of the topological parameter a to both systems by increasing the value of a from 0.05 to 0.2. The computer program in Section A.8 simulates this effect in lattices of nonidentical oscillators and the computer program in Section A.9 simulates this effect in populations of globally coupled nonidentical oscillators.

Figure 4.5 illustrates the standard deviation of frequencies $\sigma(\varepsilon)$ in a three-dimensional lattice of 200 nonidentical Rossler oscillators as a function of a . Figure 4.6 illustrates the standard deviation of frequencies $\sigma(\varepsilon)$ in a three-dimensional population of 100 globally coupled nonidentical oscillators as a function of a . In both figures phase synchronization occurs at $\sigma(\varepsilon) = 0$. From Fig. 4.5 we see that as a increases the coupling strength ε needed for phase synchronization decreases. While from Fig. 4.6 we see that as a increases the coupling strength ε needed for phase synchronization increases. It indicates the effect of the system parameter on lattices of oscillators is different from that on populations of globally coupled oscillators. The former strengthens the occurrence of phase synchronization while the later weakens the occurrence of phase synchronization.

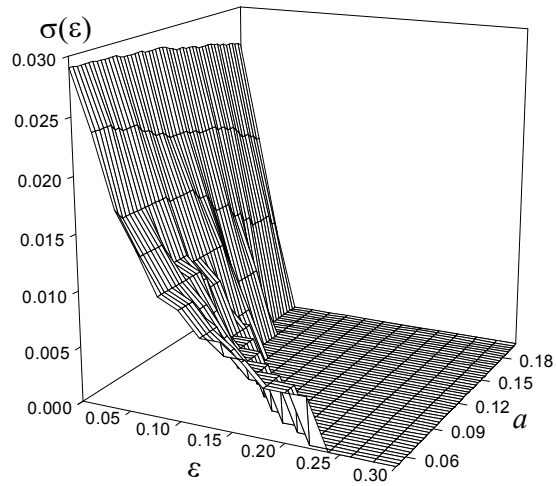


Fig. 4.5. Standard deviation of frequencies in a three-dimensional lattice of 200 nonidentical Rossler oscillators with variation of a .

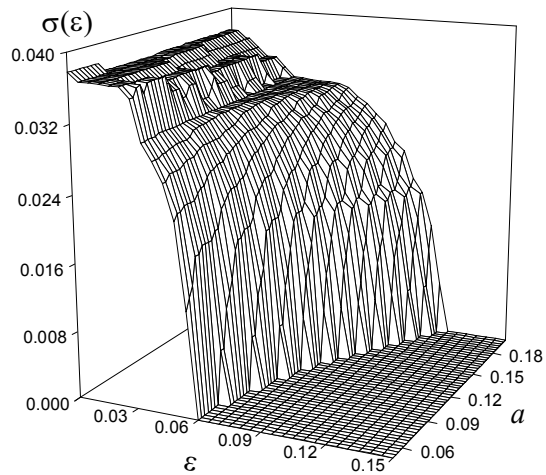


Fig. 4.6. Standard deviation of frequencies in a three-dimensional population of 100 globally coupled nonidentical Rossler oscillators with variation of a .

CHAPTER FIVE

CONCLUSION

The main purpose of this thesis is to study phase synchronization in large populations of nonidentical chaotic oscillators using the Rossler system as a prototypic model. For which the Rossler system was generalized from a single uncoupled system to the multi-dimensional N -coupled system with the general coupling components that covers different types of couplings, i.e., one oscillator with periodic external forcing, two-coupled nonidentical oscillators, lattices of nonidentical oscillators, and populations of globally coupled nonidentical oscillators. Following the generalized coupled Rossler system, the numerical simulations for phase synchronization were performed. Based on the results of the numerical simulations, some specific remarks can be drawn as follows.

1. With the increase of the topological parameter a , the trajectory of the Rossler system on the phase plane displays the transition from a limit cycle to a phase-coherent attractor, and then to a funnel attractor. The approach used in this thesis is for limit cycle and phase-coherent attractors.
2. The phase synchronization of one oscillator with periodic external forcing depends on the amplitude and frequency of external forces. One oscillator with one-dimensional external force shows a larger synchronization region than one oscillator with two- and three-dimensional external forces.
3. The phase synchronization of two-coupled nonidentical oscillators depends on the coupling strength and the frequency mismatch. Two-coupled oscillators along

the x -, y -, and z -coordinates shows a larger synchronization region than two-coupled oscillators along the x - or y -coordinate and along the x - and y -coordinates.

4. The critical values of the coupling strength, which generate the phase synchronization regions to lattices of oscillators and populations of globally coupled oscillators, are different for different coupling dimensions or population sizes.
5. As the topological parameter a increases the coupling strength ε needed for phase synchronization decreases, while as the topological parameter a increases the coupling strength ε needed for phase synchronization increases.

The current work may be extended to other types of coupling for the Rossler system and other nonlinear dynamic systems with limit cycles and phase-coherent attractors.

APPENDIX A
COMPUTER PROGRAMS FOR NUMERICAL SIMULATIONS

A.1 Trajectory of Rossler System on Phase Plane

```
/* This program computes the phase plane's trajectory of the Rossler
   system.

   Written by: Limin Qi, 03-25-2005
*/
#include <iostream.h>
#include <stdio.h>
#include <math.h>
#define NDIM 3
#define NPARAM 3
#define NSTEP 20000
#define NSKIP 5000

void rossler (double t, double param[], double x[], double dx[])
{
    double a, b, c;
    a = param[0]; b = param[1]; c = param[2];

    dx[0] = -x[1] - x[2];
    dx[1] = x[0] + a*x[1];
    dx[2] = b + x[2]*(x[0] - c);
}

void runge_kutta (double t, double dt, double param[],
                 double x[], double xnew[])
{
    double f1[NDIM], f2[NDIM], f3[NDIM], f4[NDIM], xtemp[NDIM];
    double t_half, half_dt, t_full;
    int i;

    half_dt = 0.5 * dt;
    rossler (t, param, x, f1);
    t_half = t + half_dt;
    for (i = 0; i < NDIM; i++) xtemp[i] = x[i] + half_dt*f1[i];
    rossler (t_half, param, xtemp, f2);
    for (i = 0; i < NDIM; i++) xtemp[i] = xtemp[i] + half_dt*f2[i];
    rossler (t_half, param, xtemp, f3);
    t_full = t + dt;
    for (i = 0; i < NDIM; i++) xtemp[i] = xtemp[i] + dt * f3[i];
    rossler (t_full, param, xtemp, f4);
    for (i = 0; i < NDIM; i++)
        xnew[i] = xtemp[i] + dt/6.0 * (f1[i] + f4[i] + 2.0*(f2[i] + f3[i]));
}

int main()
{
    double param[NPARAM] = {0.15, 0.4, 8.5},
           xold[NDIM], xnew[NDIM]={-1., 0., 0.};
    double t = 0.0, dt = 0.1;
    int i, j;

    FILE *fout;
    fout = fopen("ros1.txt", "w");

    for (i = 0; i < NSTEP; i++){
        for (j = 0; j < NDIM; j++) xold[j] = xnew[j];
        runge_kutta (t, dt, param, xold, xnew);
        if (i > NSKIP)
            fprintf(fout, "%8.4f %8.4f\n", xnew[0], xnew[1]);
    }
    // fprintf(fout, "%6.2f %8.4f\n", t, xnew[1]);
}
```

```

        t += dt;
    }
    fclose(fout);
    return 0;
}

```

A.2 One Oscillator with Periodic External Forcing and Fixed Topological Parameter

```

/* This program simulates phase synchronization of one Rossler oscillator
with periodic external forcing by computing the difference between the
frequency of the external force and the observed frequency of the Rossler
oscillator.

```

```

Written by: Limin Qi, 03-25-2005

```

```

*/
#include <iostream.h>
#include <stdio.h>
#include <math.h>
#define NDIM 3
#define NPARAM 3
#define NCPDIM 3
#define NE 51
#define NNU 51
#define NSTEP 20000

void rossler (double e, double nu, double t, double param[],
             double cpdim[], double x[], double dx[])
{
    double a, b, c, cx, cy, cz;
    a = param[0]; b = param[1]; c = param[2];
    cx = cpdim[0]; cy = cpdim[1]; cz = cpdim[2];

    dx[0] = -x[1] - x[2]          + cx*e*cos(nu*t);
    dx[1] = x[0] + a*x[1]        + cy*e*cos(nu*t);
    dx[2] = b + (x[0] - c)*x[2]  + cz*e*cos(nu*t);
}

void runge_kutta (double e, double nu, double t, double dt,
                 double param[], double cpdim[], double x[], double xout[])
{
    double F1[NDIM], F2[NDIM], F3[NDIM], F4[NDIM], xtemp[NDIM];
    double t_half, half_dt, t_full;
    int i;
    half_dt = 0.5 * dt;
    rossler (e, nu, t, param, cpdim, x, F1);
    t_half = t + half_dt;
    for (i = 0; i < NDIM; i++)
        xtemp[i] = x[i] + half_dt*F1[i];
    rossler (e, nu, t_half, param, cpdim, xtemp, F2);
    for (i = 0; i < NDIM; i++)
        xtemp[i] = x[i] + half_dt*F2[i];
    rossler (e, nu, t_half, param, cpdim, xtemp, F3);
    t_full = t + dt;
    for (i = 0; i < NDIM; i++)
        xtemp[i] = x[i] + dt * F3[i];
    rossler (e, nu, t_full, param, cpdim, xtemp, F4);
    for (i = 0; i < NDIM; i++)
        xout[i] = x[i] + dt/6.0 *
                (F1[i] + F4[i] + 2.0*(F2[i] + F3[i]));
}

```

```

int main()
{
    double param[NPARAM] = {0.15, 0.4, 8.5};
    double cpdim[NCPDIM] = {1., 1., 1.};
    double xin[NDIM], xout0[NDIM], xout1[NDIM], xout2[NDIM];
    double dsigmax, dsigmay, nx[NE], ny[NE];
    double e, de = 0.02, nu, dnu = 0.005, t, dt = 0.1;
    int i, j, k, l;

    FILE *fout;
        fout = fopen("rosEfxyz.txt", "w");

    nu = 0.9;
    for(i = 0; i < NNU; i++){
        e = 0.;
        for(j = 0; j < NE; j++){
            nx[j] = 0.;
            ny[j] = 0.;
            t = 0.;
            xout0[0] = -1.;
            xout0[1] = 0.;
            xout0[2] = 0.;
            t = dt;
            for(k = 0; k < NDIM; k++) xin[k] = xout0[k];
            runge_kutta (e, nu, t, dt, param, cpdim, xin, xout1);
            t = 2.*dt;
            for(l = 2; l < NSTEP + 1; l++){
                for(k = 0; k < NDIM; k++) xin[k] = xout1[k];
                runge_kutta (e, nu, t, dt, param, cpdim, xin, xout2);
                if(xout0[0] < xout1[0] && xout1[0] > xout2[0])
                    nx[j] += 1.;
                if(xout0[1] < xout1[1] && xout1[1] > xout2[1])
                    ny[j] += 1.;

                for(k = 0; k < NDIM; k++){
                    xout0[k] = xout1[k];
                    xout1[k] = xout2[k];
                }
                t += dt;
            }
            dsigmax = 2.*3.14159*(double)nx[j]/(double)NSTEP/dt - nu;
//          fprintf(fout, " %8.4f %8.4f %8.4f\n", nu, e, dsigmax);

            dsigmay = 2.*3.14159*(double)ny[j]/(double)NSTEP/dt - nu;
            fprintf(fout, " %8.4f %8.4f %8.4f\n", nu, e, dsigmay);
            e += de;
        }
        nu += dnu;
    }
    fclose(fout);
    return 0;
}

```

A.3 Two-Coupled Nonidentical Oscillators with Fixed Topological Parameter

```

/* This program simulates phase synchronization of two-coupled Rossler
oscillators by computing the difference between two observed frquencies.

```

```

Written by: Limin Qi, 03-25-2005
*/

#include <iostream.h>
#include <stdio.h>
#include <math.h>
#define NDIM 6
#define NPARAM 3
#define NPCDIM 3
#define NEPS 51
#define NOMEGA 51
#define NSTEP 20000

void rossler (double eps, double domega, double t, double param[],
             double cpdim[], double x[], double dx[])
{
    double a, b, c, cx, cy, cz;
    a = param[0]; b = param[1]; c = param[2];
    cx = cpdim[0]; cy = cpdim[1]; cz = cpdim[2];

    dx[0] = -(1. + domega)*x[1] - x[2] + cx*eps*(x[3] - x[0]);
    dx[1] = (1. + domega)*x[0] + a*x[1] + cy*eps*(x[4] - x[1]);
    dx[2] = b + x[2]*(x[0] - c) + cz*eps*(x[5] - x[2]);
    dx[3] = -(1. - domega)*x[4] - x[5] + cx*eps*(x[0] - x[3]);
    dx[4] = (1. - domega)*x[3] + a*x[4] + cy*eps*(x[1] - x[4]);
    dx[5] = b + x[5]*(x[3] - c) + cz*eps*(x[2] - x[5]);
}

void runge_kutta (double eps, double domega, double t, double dt,
                double param[], double cpdim[], double x[], double xout[])
{
    double F1[NDIM], F2[NDIM], F3[NDIM], F4[NDIM], xtemp[NDIM];
    double t_half, half_dt, t_full;
    int i;

    half_dt = 0.5 * dt;
    rossler (eps, domega, t, param, cpdim, x, F1);
    t_half = t + half_dt;
    for (i = 0; i < NDIM; i++)
        xtemp[i] = x[i] + half_dt*F1[i];
    rossler (eps, domega, t_half, param, cpdim, xtemp, F2);
    for (i = 0; i < NDIM; i++)
        xtemp[i] = x[i] + half_dt*F2[i];
    rossler (eps, domega, t_half, param, cpdim, xtemp, F3);
    t_full = t + dt;
    for (i = 0; i < NDIM; i++)
        xtemp[i] = x[i] + dt * F3[i];
    rossler (eps, domega, t_full, param, cpdim, xtemp, F4);
    for (i = 0; i < NDIM; i++)
        xout[i] = x[i] + dt/6.0 *
            (F1[i] + F4[i] + 2.0*(F2[i] + F3[i]));
}

int main()
{
    double param[NPARAM] = {0.15, 0.4, 8.5};
    double cpdim[NPCDIM] = {1., 1., 1.};
    double xin[NDIM], xout0[NDIM], xout1[NDIM], xout2[NDIM];
    double nx1[NEPS], nx2[NEPS], ny1[NEPS], ny2[NEPS];
    double eps, depts = 0.004, t, dt = 0.1, domegax, domegay;
    double domega, ddomega = 0.002;
    int i, j, k, l;

    FILE *fout;
    fout = fopen("ros21xyz.txt", "w");
}

```

```

domega = 0.;
for(i = 0; i < NOMEGA; i++){
    eps = 0.;
    for(j = 0; j < NEPS; j++){
        nx1[j] = 0.;
        nx2[j] = 0.;
        ny1[j] = 0.;
        ny2[j] = 0.;
        t = 0.;
        xout0[0] = -1.;
        xout0[1] = 0.;
        xout0[2] = 0.;
        xout0[3] = -1.;
        xout0[4] = 0.;
        xout0[5] = 0.;
        t = dt;
        for(k = 0; k < NDIM; k++) xin[k] = xout0[k];
        runge_kutta (eps, domega, t, dt, param, cpdim, xin,
                    xout1);

        t = 2.*dt;
        for(l = 2; l < NSTEP + 1; l++){
            for(k = 0; k < NDIM; k++) xin[k] = xout1[k];
            runge_kutta (eps, domega, t, dt, param, cpdim, xin,
                        xout2);
            if(xout0[0] < xout1[0] && xout1[0] > xout2[0])
                nx1[j] += 1.;
            if(xout0[3] < xout1[3] && xout1[3] > xout2[3])
                nx2[j] += 1.;
            if(xout0[1] < xout1[1] && xout1[1] > xout2[1])
                ny1[j] += 1.;
            if(xout0[4] < xout1[4] && xout1[4] > xout2[4])
                ny2[j] += 1.;

            for(k = 0; k < NDIM; k++){
                xout0[k] = xout1[k];
                xout1[k] = xout2[k];
            }
            t += dt;
        }
        domegax = 2.*3.14159*(nx1[j] - nx2[j])/(double)NSTEP/dt;
        fprintf(fout, " %8.4f %8.4f %8.4f\n", domega, eps, domegax);

        domegay = 2.*3.14159*(ny1[j] - ny2[j])/(double)NSTEP/dt;
        fprintf(fout, " %8.4f %8.4f %8.4f\n", domega, eps, domegay);
        eps += depts;
    }
    domega += ddomega;
}
fclose(fout);
return 0;
}

```

A.4 One Oscillator with Periodic External Forces and Variation of Topological Parameter

```

/* This program simulates phase synchronization of one Rossler oscillator
with periodic external forcing by computing the difference between the
frequency of the external force and the observed frequency of the Rossler

```

oscillator with variation of system parameter a.

```
Written by: Limin Qi, 03-25-2005
*/

#include <iostream.h>
#include <stdio.h>
#include <math.h>
#define NDIM 3
#define NPARAM 2
#define NCPDIM 3
#define NA 51
#define NNU 51
#define NSTEP 20000

void rossler (double a, double e, double nu, double t, double param[],
             double cpdim[], double x[], double dx[])
{
    double b, c, cx, cy, cz;
    b = param[0]; c = param[1];
    cx = cpdim[0]; cy = cpdim[1]; cz = cpdim[2];

    dx[0] = -x[1] - x[2] + cx*e*cos(nu*t);
    dx[1] = x[0] + a*x[1] + cy*e*cos(nu*t);
    dx[2] = b + (x[0] - c)*x[2] + cz*e*cos(nu*t);
}

void runge_kutta (double a, double e, double nu, double t, double dt,
                 double param[], double cpdim[], double x[], double xout[])
{
    double F1[NDIM], F2[NDIM], F3[NDIM], F4[NDIM], xtemp[NDIM];
    double t_half, half_dt, t_full;
    int i;
    half_dt = 0.5 * dt;
    rossler (a, e, nu, t, param, cpdim, x, F1);
    t_half = t + half_dt;
    for (i = 0; i < NDIM; i++)
        xtemp[i] = x[i] + half_dt*F1[i];
    rossler (a, e, nu, t_half, param, cpdim, xtemp, F2);
    for (i = 0; i < NDIM; i++)
        xtemp[i] = x[i] + half_dt*F2[i];
    rossler (a, e, nu, t_half, param, cpdim, xtemp, F3);
    t_full = t + dt;
    for (i = 0; i < NDIM; i++)
        xtemp[i] = x[i] + dt * F3[i];
    rossler (a, e, nu, t_full, param, cpdim, xtemp, F4);
    for (i = 0; i < NDIM; i++)
        xout[i] = x[i] + dt/6.0 *
                (F1[i] + F4[i] + 2.0*(F2[i] + F3[i]));
}

int main()
{
    double a, param[NPARAM] = {0.4, 8.5};
    double cpdim[NCPDIM] = {1., 1., 1.};
    double xin[NDIM], xout0[NDIM], xout1[NDIM], xout2[NDIM];
    double dsigmax, dsigmay, nx[NA], ny[NA];
    double e = 0.5, nu, da = 0.003, dnu = 0.005, t, dt = 0.1;
    int i, j, k, l;

    FILE *fout;
    fout = fopen("rosEaxyz.txt", "w");

    nu = 0.9;
    for(i = 0; i < NNU; i++){
        a = 0.05;
```



```

for(j = 0; j < NA; j++){
    nx[j] = 0.;
    ny[j] = 0.;
    t = 0.;
    xout0[0] = -1.;
    xout0[1] = 0.;
    xout0[2] = 0.;
    t = dt;
    for(k = 0; k < NDIM; k++) xin[k] = xout0[k];
    runge_kutta (a, e, nu, t, dt, param, cpdim, xin, xout1);
    t = 2.*dt;
    for(l = 2; l < NSTEP + 1; l++){
        for(k = 0; k < NDIM; k++) xin[k] = xout1[k];
        runge_kutta (a, e, nu, t, dt, param, cpdim, xin, xout2);
        if(xout0[0] < xout1[0] && xout1[0] > xout2[0])
            nx[j] += 1.;
        if(xout0[1] < xout1[1] && xout1[1] > xout2[1])
            ny[j] += 1.;

        for(k = 0; k < NDIM; k++){
            xout0[k] = xout1[k];
            xout1[k] = xout2[k];
        }
        t += dt;
    }
    dsigmax = 2.*3.14159*(double)nx[j]/(double)NSTEP/dt - nu;
//    fprintf(fout, " %8.4f %8.4f %8.4f\n", nu, e, dsigmax);

    dsigmay = 2.*3.14159*(double)ny[j]/(double)NSTEP/dt - nu;
    fprintf(fout, " %8.4f %8.4f %8.4f\n", nu, a, dsigmay);
    a += da;
}
nu += dnu;
}
fclose(fout);
return 0;
}

```

A.5 Two-Coupled Nonidentical Oscillators with Variation of Topological Parameter

/* This program simulates phase synchronization of two-coupled Rossler oscillators by computing the difference between two observed frquencies. with variation of system parameter a.

```

Written by: Limin Qi, 03-25-2005
*/
#include <iostream.h>
#include <stdio.h>
#include <math.h>
#define NDIM 6
#define NPARAM 2
#define NCPDIM 3
#define NEPS 51
#define NA 51
#define NSTEP 20000

void rossler (double a, double eps, double t, double param[],
             double cpdim[], double x[], double dx[])
{
    double b, c, cx, cy, cz, omegal = 1.015, omega2 = 0.985;
    b = param[0]; c = param[1];

```

```

cx = cpdim[0]; cy = cpdim[1]; cz = cpdim[2];

dx[0] = -omegal*x[1] - x[2] + cx*eps*(x[3] - x[0]);
dx[1] = omegal*x[0] + a*x[1] + cy*eps*(x[4] - x[1]);
dx[2] = b + x[2]*(x[0] - c) + cz*eps*(x[5] - x[2]);
dx[3] = -omega2*x[4] - x[5] + cx*eps*(x[0] - x[3]);
dx[4] = omega2*x[3] + a*x[4] + cy*eps*(x[1] - x[4]);
dx[5] = b + x[5]*(x[3] - c) + cz*eps*(x[2] - x[5]);
}

void runge_kutta (double a, double eps, double t, double dt,
double param[], double cpdim[], double x[], double xout[])
{
double F1[NDIM], F2[NDIM], F3[NDIM], F4[NDIM], xtemp[NDIM];
double t_half, half_dt, t_full;
int i;

half_dt = 0.5 * dt;
rossler (a, eps, t, param, cpdim, x, F1);
t_half = t + half_dt;
for (i = 0; i < NDIM; i++)
xtemp[i] = x[i] + half_dt*F1[i];
rossler (a, eps, t_half, param, cpdim, xtemp, F2);
for (i = 0; i < NDIM; i++)
xtemp[i] = x[i] + half_dt*F2[i];
rossler (a, eps, t_half, param, cpdim, xtemp, F3);
t_full = t + dt;
for (i = 0; i < NDIM; i++)
xtemp[i] = x[i] + dt * F3[i];
rossler (a, eps, t_full, param, cpdim, xtemp, F4);
for (i = 0; i < NDIM; i++)
xout[i] = x[i] + dt/6.0 *
(F1[i] + F4[i] + 2.0*(F2[i] + F3[i]));
}

int main()
{
double a, param[NPARAM] = {0.4, 8.5};
double cpdim[NCPDIM] = {1., 1., 1.};
double xin[NDIM], xout0[NDIM], xout1[NDIM], xout2[NDIM];
double nx1[NEPS], nx2[NEPS], ny1[NEPS], ny2[NEPS], domegax, domegay;
double eps, depx = 0.004, da = 0.003, t, dt = 0.1;
int i, j, k, l;

FILE *fout;
fout = fopen("ros21axyz.txt", "w");

a = 0.05;
for(i = 0; i < NA; i++){
eps = 0.;
for(j = 0; j < NEPS; j++){
nx1[j] = 0.;
nx2[j] = 0.;
ny1[j] = 0.;
ny2[j] = 0.;
t = 0.;
xout0[0] = -1.;
xout0[1] = 0.;
xout0[2] = 0.;
xout0[3] = -1.;
xout0[4] = 0.;
xout0[5] = 0.;
t = dt;
for(k = 0; k < NDIM; k++) xin[k] = xout0[k];
runge_kutta (a, eps, t, dt, param, cpdim, xin, xout1);
}
}
}

```

```

t = 2.*dt;
for(l = 2; l < NSTEP + 1; l++){
    for(k = 0; k < NDIM; k++){
        xin[k] = xout1[k];
        runge_kutta (a, eps, t, dt, param, cpdim, xin, xout2);
        if(xout0[0] < xout1[0] && xout1[0] > xout2[0])
            nx1[j] += 1.;
        if(xout0[3] < xout1[3] && xout1[3] > xout2[3])
            nx2[j] += 1.;
        if(xout0[1] < xout1[1] && xout1[1] > xout2[1])
            ny1[j] += 1.;
        if(xout0[4] < xout1[4] && xout1[4] > xout2[4])
            ny2[j] += 1.;

        for(k = 0; k < NDIM; k++){
            xout0[k] = xout1[k];
            xout1[k] = xout2[k];
        }
        t += dt;
    }
    domegax = 2.*3.14159*(nx1[j] - nx2[j])/(double)NSTEP/dt;
    fprintf(fout, " %8.4f %8.4f %8.4f\n", a, eps, domegax);

    domegay = 2.*3.14159*(ny1[j] - ny2[j])/(double)NSTEP/dt;
    fprintf(fout, " %8.4f %8.4f %8.4f\n", a, eps, domegay);
    eps += depts;
}
a += da;
}
fclose(fout);
return 0;
}

```

A.6 Lattices of Nonidentical Oscillators with Fixed Topological Parameter

```

/* This program simulates phase synchronization in three-dimensional
lattice of nonidentical Rossler oscillators by computing the
standard deviation of the observed frequencies.

Written by: Limin Qi, 03-25-2005
*/

#include <iostream.h>
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#define NPARAM 3
#define NCPDIM 3
#define NSYS 1000
#define NSTEP 1000
#define NEPS 100

void rossler (double eps, double t, double om[], double param[],
double cpdim[], double x[NSYS+1], double y[NSYS+1], double z[NSYS+1],
double dx[], double dy[], double dz[])
{
    double a, b, c, cx, cy, cz;
    double latticex[NSYS], latticey[NSYS], latticez[NSYS];

    a = param[0]; b = param[1]; c = param[2];
    cx = cpdim[0]; cy = dcpdim[1]; cz = cpdim[2];

```

```

        x[0] = x[1];
    y[0] = y[1];
        z[0] = z[1];
        x[NSYS] = x[NSYS-1];
    y[NSYS] = y[NSYS-1];
    z[NSYS] = z[NSYS-1];

    for (int i = 1; i < NSYS; i++) {

        latticex[i] = x[i+1] - 2.*x[i] + x[i-1];
        latticey[i] = y[i+1] - 2.*y[i] + y[i-1];
        latticez[i] = z[i+1] - 2.*z[i] + z[i-1];

        dx[i] = -om[i]*y[i] - z[i] + cx*eps*latticex[i];
        dy[i] = om[i]*x[i] + a*y[i] + cy*eps*latticey[i];
        dz[i] = b + z[i]*(x[i] - c) + cz*eps*latticez[i];
    }
}

void runge_kutta (double eps, double t, double dt, double param[],
    double cpdim[], double om[NCPDIM], double x[NSYS+1], double y[NSYS+1],
    double z[NSYS+1],
    double xnew[NSYS+1], double ynew[NSYS+1], double znew[NSYS+1])
{
    double f11[NSYS], f12[NSYS], f13[NSYS];
    double f21[NSYS], f22[NSYS], f23[NSYS];
    double f31[NSYS], f32[NSYS], f33[NSYS];
    double f41[NSYS], f42[NSYS], f43[NSYS];
    double xtem[NSYS+1], ytem[NSYS+1], ztem[NSYS+1];
    double t_half, half_dt, t_full;
    int i;

    half_dt = 0.5 * dt;
    rossler (eps, t, om, param, cpdim, x, y, z, f11, f12, f13);

    t_half = t + half_dt;
    for (i = 0; i < NSYS; i++) {
        xtem[i] = x[i] + half_dt*f11[i];
        ytem[i] = y[i] + half_dt*f12[i];
        ztem[i] = z[i] + half_dt*f13[i];
    }
    rossler (eps, t_half, om, param, cpdim, xtem, ytem, ztem, f21, f22, f23);

    for (i = 0; i < NSYS; i++) {
        xtem[i] = x[i] + half_dt*f21[i];
        ytem[i] = y[i] + half_dt*f22[i];
        ztem[i] = z[i] + half_dt*f23[i];
    }
    rossler (eps, t_half, om, param, cpdim, xtem, ytem, ztem, f31, f32, f33);

    t_full = t + dt;
    for (i = 0; i < NSYS; i++) {
        xtem[i] = x[i] + dt*f31[i];
        ytem[i] = y[i] + dt*f31[i];
        ztem[i] = z[i] + dt*f31[i];
    }
    rossler (eps, t_full, om, param, cpdim, xtem, ytem, ztem, f41, f42, f43);

    for (i = 0; i < NSYS; i++) {
        xnew[i] = x[i]+dt/6.0*(f11[i]+f41[i]+2.0*(f21[i]+f31[i]));
        ynew[i] = y[i]+dt/6.0*(f12[i]+f42[i]+2.0*(f22[i]+f32[i]));
        znew[i] = z[i]+dt/6.0*(f13[i]+f43[i]+2.0*(f23[i]+f33[i]));
    }
}

```

```

int main()
{
    int    i, j, k;
    double xx[NSYS+1], yy[NSYS+1], zz[NSYS+1];
    double x0[NSYS+1], y0[NSYS+1], z0[NSYS+1];
    double x1[NSYS+1], y1[NSYS+1], z1[NSYS+1];
    double x2[NSYS+1], y2[NSYS+1], z2[NSYS+1];
    double param[NPARAM] = {0.15, 0.4, 8.5}, om[NSYS];
    double cpdim[NCPDIM] = {1., 1., 1.};
    double eps, deps = 0.01, t, dt = 0.1;
    double nx[NSYS], mfg_x, sumx, sumxx, sd_frqx;
    double ny[NSYS], mfg_y, sumy, sumyy, sd_frqy;
    unsigned seed;

    FILE *fout;
        fout = fopen("rLf1000.txt", "w");

    printf ("Enter an integer randomly: ");
    scanf ("%u", &seed);
    srand(seed);

    for (i = 0; i < NSYS; i++) {
        om[i] = 0.9450 + 0.000015*(1 + (rand() % 5000));
        printf ( " om %8.4f\n", om[i]);
    }

    eps = 0.0;
    for (k = 0; k < NEPS; k++) {

        t = 0.;
        for (i = 0; i < NSYS; i++) {
            nx[i] = 0.;
            ny[i] = 0.;
            x0[i] = -1.;
            y0[i] = 0.;
            z0[i] = 0.;
        }
        t = dt;
        for (i = 0; i < NSYS; i++) {
            xx[i] = x0[i];
            yy[i] = y0[i];
            zz[i] = z0[i];
        }

        runge_kutta (eps, t, dt, param, cpdim, om, xx, yy, zz, x1, y1, z1);

        t = 2.*dt;
        for (j = 2; j < NSTEP; j++) {

            for (i = 0; i < NSYS; i++) {
                xx[i] = x1[i];
                yy[i] = y1[i];
                zz[i] = z1[i];
            }

            runge_kutta (eps, t, dt, param, cpdim, om, xx, yy, zz, x2, y2, z2);

            for (i = 0; i < NSYS; i++) {
                if (x0[i] < x1[i] && x1[i] > x2[i]) nx[i] += 1.;
                if (y0[i] < y1[i] && y1[i] > y2[i]) ny[i] += 1.;
                x0[i] = x1[i];
                x1[i] = x2[i];
                y0[i] = y1[i];
            }
        }
    }
}

```

```

        y1[i] = y2[i];
        }
        t += dt;
    }

    sumx = 0.;
    sumy = 0.;
    for (i = 0; i < NSYS; i++) {
        sumx += 2.*3.14159*nx[i]/(double)NSTEP/dt;
        sumy += 2.*3.14159*ny[i]/(double)NSTEP/dt;
    }

    mfq_x = sumx/(double)NSYS;
    mfq_y = sumy/(double)NSYS;

    sumxx = 0.;
    sumyy = 0.;
    for (i = 0; i < NSYS; i++) {
        sumxx += pow(2.*3.14159*nx[i]/(double)NSTEP/dt - mfq_x, 2);
        sumyy += pow(2.*3.14159*ny[i]/(double)NSTEP/dt - mfq_y, 2);
    }

    sd_frqx = sqrt(sumxx/((double)NSYS-1.));
    sd_frqy = sqrt(sumyy/((double)NSYS-1.));

//    fprintf(fout, " %6.4f %10.7f\n", eps, sd_frqx);
//    fprintf(fout, " %10.7f\n", sd_frqy);
//    fprintf(fout, " %6.4f %10.7f\n", eps, sd_frqy);
    eps += deps;
    cout << eps << " " << sd_frqx << endl;
//    cout << eps << " " << sd_frqy << endl;
}
fclose(fout);
return 0;
}

```

A.7 Populations of Globally Coupled Nonidentical Oscillators with Fixed Topological Parameter

```

/* This program simulates phase synchronization in three-dimensional
globally coupled population of nonidentical Rossler oscillators
by computing the standard deviation of the observed frequencies.

```

```

Written by: Limin Qi, 03-25-2005

```

```

*/

```

```

#include <iostream.h>
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#define NPARAM 3
#define NCPDIM 3
#define NSYS 500
#define NSTEP 1000
#define NEPS 31

```

```

void rossler (double eps, double t, double om[], double param[],
double cpdim[], double sumx[], double sumy[], double sumz[],
double x[], double y[], double z[],

```

```

    double dx[], double dy[], double dz[])
{
    double a, b, c, cx, cy, cz;

    a = param[0]; b = param[1]; c = param[2];
    cx = cpdim[0]; cy = cpdim[1]; cz = cpdim[2];

    for (int i = 0; i < NSYS; i++) {
        for (int j = 0; j < NSYS; j++) {
            sumx[i] += x[j] - x[i];
            sumy[i] += y[j] - y[i];
            sumz[i] += z[j] - z[i];
        }
        dx[i] = -om[i]*y[i] - z[i] + cx*eps*sumx[i]/((double)NSYS - 1.);
        dy[i] = om[i]*x[i] + a*y[i] + cy*eps*sumy[i]/((double)NSYS - 1.);
        dz[i] = b + z[i]*(x[i] - c) + cz*eps*sumz[i]/((double)NSYS - 1.);
    }
}

void runge_kutta (double eps, double t, double dt, double param[],
    double cpdim[], double om[], double x[], double y[], double z[],
    double xnew[], double ynew[], double znew[])
{
    double f11[NSYS], f12[NSYS], f13[NSYS];
    double f21[NSYS], f22[NSYS], f23[NSYS];
    double f31[NSYS], f32[NSYS], f33[NSYS];
    double f41[NSYS], f42[NSYS], f43[NSYS];
    double xtem[NSYS], ytem[NSYS], ztem[NSYS];
    double sumx[NSYS], sumy[NSYS], sumz[NSYS];
    double t_half, half_dt, t_full;
    int i;

    for (i = 0; i < NSYS; i++) {
        sumx[i] = 0.;
        sumy[i] = 0.;
        sumz[i] = 0.;
    }

    half_dt = 0.5 * dt;
    rossler (eps, t, om, param, cpdim, sumx, sumy, sumz, x, y, z,
        f11, f12, f13);

    t_half = t + half_dt;
    for (i = 0; i < NSYS; i++) {
        xtem[i] = x[i] + half_dt*f11[i];
        ytem[i] = y[i] + half_dt*f12[i];
        ztem[i] = z[i] + half_dt*f13[i];
    }
    rossler (eps, t_half, om, param, cpdim, sumx, sumy, sumz, xtem, ytem, ztem,
        f21, f22, f23);

    for (i = 0; i < NSYS; i++) {
        xtem[i] = x[i] + half_dt*f21[i];
        ytem[i] = y[i] + half_dt*f22[i];
        ztem[i] = z[i] + half_dt*f23[i];
    }
    rossler (eps, t_half, om, param, cpdim, sumx, sumy, sumz, xtem, ytem, ztem,
        f31, f32, f33);

    t_full = t + dt;
    for (i = 0; i < NSYS; i++) {
        xtem[i] = x[i] + dt*f31[i];
        ytem[i] = y[i] + dt*f31[i];
        ztem[i] = z[i] + dt*f31[i];
    }
}

```

```

rossler (eps, t_full, om, param, cpdim, sumx, sumy, sumz, xtem, ytem, ztem,
        f41, f42, f43);

    for (i = 0; i < NSYS; i++) {
        xnew[i] = x[i]+dt/6.0*(f11[i]+f41[i]+2.0*(f21[i]+f31[i]));
        ynew[i] = y[i]+dt/6.0*(f12[i]+f42[i]+2.0*(f22[i]+f32[i]));
        znew[i] = z[i]+dt/6.0*(f13[i]+f43[i]+2.0*(f23[i]+f33[i]));
    }
}

int main()
{
    int    i, j, k;
    double xx[NSYS], yy[NSYS], zz[NSYS];
    double x0[NSYS], y0[NSYS], z0[NSYS];
    double x1[NSYS], y1[NSYS], z1[NSYS];
    double x2[NSYS], y2[NSYS], z2[NSYS];
    double param[NPARAM] = {0.15, 0.4, 8.5}, om[NSYS];
    double cpdim[NCPDIM] = {1., 1., 1.};
    double eps, deps = 0.005, t, dt = 0.1;
    double nx[NSYS], mfg_x, sumx, sumxx, sd_frqx;
    double ny[NSYS], mfg_y, sumy, sumyy, sd_frqy;
    unsigned seed;

    FILE *fout;
        fout = fopen("rNf50.txt", "w");

    printf ("Enter a positive integer randomly: ");
    scanf ("%u", &seed);
    srand(seed);

    for (i = 0; i < NSYS; i++) {
        om[i] = 0.9450 + 0.000025*(1 + (rand() % 5000));
        printf ( " om %8.4f\n", om[i]);
    }

    eps = 0.0;
    for (k = 0; k < NEPS; k++) {

        t = 0.;
        for (i = 0; i < NSYS; i++) {
            nx[i] = 0.;
            ny[i] = 0.;
            x0[i] = -1.;
            y0[i] = 0.;
            z0[i] = 0.;
        }
        t = dt;
        for (i = 0; i < NSYS; i++) {
            xx[i] = x0[i];
            yy[i] = y0[i];
            zz[i] = z0[i];
        }

        runge_kutta (eps, t, dt, param, cpdim, om, xx, yy, zz, x1, y1, z1);

        t = 2.*dt;
        for (j = 2; j < NSTEP; j++) {

            for (i = 0; i < NSYS; i++) {
                xx[i] = x1[i];
                yy[i] = y1[i];
                zz[i] = z1[i];
            }
        }
    }
}

```



```

runge_kutta (eps, t, dt, param, cpdim, om, xx, yy, zz, x2, y2, z2);

    for (i = 0; i < NSYS; i++) {
        if (x0[i] < x1[i] && x1[i] > x2[i]) nx[i] += 1.;
        if (y0[i] < y1[i] && y1[i] > y2[i]) ny[i] += 1.;
        x0[i] = x1[i];
        x1[i] = x2[i];
        y0[i] = y1[i];
        y1[i] = y2[i];
    }
    t += dt;
}

sumx = 0.;
sumy = 0.;
for (i = 0; i < NSYS; i++) {
    sumx += 2.*3.14159*nx[i]/(double)NSTEP/dt;
    sumy += 2.*3.14159*ny[i]/(double)NSTEP/dt;
}

mfq_x = sumx/(double)NSYS;
mfq_y = sumy/(double)NSYS;

sumxx = 0.;
sumyy = 0.;
for (i = 0; i < NSYS; i++) {
    sumxx += pow(2.*3.14159*nx[i]/(double)NSTEP/dt - mfq_x, 2);
    sumyy += pow(2.*3.14159*ny[i]/(double)NSTEP/dt - mfq_y, 2);
}

sd_frqx = sqrt(sumxx/((double)NSYS-1.));
sd_frqy = sqrt(sumyy/((double)NSYS-1.));

fprintf(fout, " %10.7f\n", sd_frqy);
//
// fprintf(fout, " %6.4f %10.7f\n", eps, sd_frqx);
// fprintf(fout, " %6.4f %10.7f\n", eps, sd_frqy);
eps += deps;
// cout << eps << " " << sd_frqx << endl;
cout << eps << " " << sd_frqy << endl;
}
fclose(fout);
return 0;
}

```

A.8 Lattices of Nonidentical Oscillators with Variation of Topological Parameter

```

/* This program simulates phase synchronization in three-dimensional
lattice of nonidentical Rossler oscillators by computing the
standard deviation of the observed frequencies with variation
of system parameter a.

```

Written by: Limin Qi, 03-25-2005

```
*/
```

```

#include <iostream.h>
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

```

```

#define NPARAM 2
#define NCPDIM 3
#define NSYS 1000
#define NSTEP 1000
#define NEPS 51
#define NA 51

void rossler (double a, double eps, double t, double om[], double param[],
             double cpdim[], double x[NSYS+1], double y[NSYS+1], double z[NSYS+1],
             double dx[], double dy[], double dz[])
{
    double b, c, cx, cy, cz,
          double latticex[NSYS], latticey[NSYS], latticexz[NSYS];

    b = param[0]; c = param[1];
    cx = cpdim[0]; cy = cpdim[1]; cz = cpdim[2];

    x[0] = x[1];
    y[0] = y[1];
    z[0] = z[1];
    x[NSYS] = x[NSYS-1];
    y[NSYS] = y[NSYS-1];
    z[NSYS] = z[NSYS-1];

    for (int i = 1; i < NSYS; i++) {

        latticex[i] = x[i+1] - 2.*x[i] + x[i-1];
        latticey[i] = y[i+1] - 2.*y[i] + y[i-1];
        latticexz[i] = z[i+1] - 2.*z[i] + z[i-1];

        dx[i] = -om[i]*y[i] - z[i] + cx*eps*latticex[i];
        dy[i] = om[i]*x[i] + a*y[i] + cy*eps*latticey[i];
        dz[i] = b + z[i]*(x[i] - c) + cz*eps*latticexz[i];
    }
}

void runge_kutta (double a, double eps, double t, double dt, double param[],
                double cpdim[], double om[],
                double x[NSYS+1], double y[NSYS+1], double z[NSYS+1],
                double xnew[NSYS+1], double ynew[NSYS+1], double znew[NSYS+1])
{
    double f11[NSYS], f12[NSYS], f13[NSYS];
    double f21[NSYS], f22[NSYS], f23[NSYS];
    double f31[NSYS], f32[NSYS], f33[NSYS];
    double f41[NSYS], f42[NSYS], f43[NSYS];
    double xtem[NSYS+1], ytem[NSYS+1], ztem[NSYS+1];
    double t_half, half_dt, t_full;
    int i;

    half_dt = 0.5 * dt;
    rossler (a, eps, t, om, param, cpdim, x, y, z, f11, f12, f13);

    t_half = t + half_dt;
    for (i = 0; i < NSYS; i++) {
        xtem[i] = x[i] + half_dt*f11[i];
        ytem[i] = y[i] + half_dt*f12[i];
        ztem[i] = z[i] + half_dt*f13[i];
    }
    rossler (a, eps, t_half, om, param, cpdim, xtem, ytem, ztem, f21, f22, f23);

    for (i = 0; i < NSYS; i++) {
        xtem[i] = x[i] + half_dt*f21[i];
        ytem[i] = y[i] + half_dt*f22[i];
        ztem[i] = z[i] + half_dt*f23[i];
    }
}

```

```

rossler (a, eps, t_half, om, param, cpdim, xtem, ytem, ztem, f31, f32, f33);

    t_full = t + dt;
    for (i = 0; i < NSYS; i++) {
        xtem[i] = x[i] + dt*f31[i];
        ytem[i] = y[i] + dt*f31[i];
        ztem[i] = z[i] + dt*f31[i];
    }
rossler (a, eps, t_full, om, param, cpdim, xtem, ytem, ztem, f41, f42, f43);

    for (i = 0; i < NSYS; i++) {
        xnew[i] = x[i]+dt/6.0*(f11[i]+f41[i]+2.0*(f21[i]+f31[i]));
        ynew[i] = y[i]+dt/6.0*(f12[i]+f42[i]+2.0*(f22[i]+f32[i]));
        znew[i] = z[i]+dt/6.0*(f13[i]+f43[i]+2.0*(f23[i]+f33[i]));
    }
}

int main()
{
    int    i, j, k, l;
    double xx[NSYS+1], yy[NSYS+1], zz[NSYS+1];
    double x0[NSYS+1], y0[NSYS+1], z0[NSYS+1];
    double x1[NSYS+1], y1[NSYS+1], z1[NSYS+1];
    double x2[NSYS+1], y2[NSYS+1], z2[NSYS+1];
    double param[NPARAM] = {0.4, 8.5}, om[NSYS];
    double cpdim[NCPDIM] = {1., 1., 1.};
    double a, da = 0.003, eps, depts = 0.02, t, dt = 0.1;
    double nx[NSYS], mfg_x, sumx, sumxx, sd_frqx;
    double ny[NSYS], mfg_y, sumy, sumyy, sd_frqy;
    unsigned seed;

    FILE *fout;
        fout = fopen("rLa.txt", "w");

    printf ("Enter an integer randomly: ");
    scanf ("%u", &seed);
    srand(seed);

    for (i = 0; i < NSYS; i++) {
        om[i] = 0.9450 + 0.000015*(1 + (rand() % 5000));
        printf ( " om %8.4f\n", om[i]);
    }

    a = 0.05;
    for (l = 0; l < NA; l++) {
        eps = 0.0;
        for (k = 0; k < NEPS; k++) {

            t = 0.;
            for (i = 0; i < NSYS; i++) {
                nx[i] = 0.;
                ny[i] = 0.;
                x0[i] = -1.;
                y0[i] = 0.;
                z0[i] = 0.;
            }
            t = dt;
            for (i = 0; i < NSYS; i++) {
                xx[i] = x0[i];
                yy[i] = y0[i];
                zz[i] = z0[i];
            }

            runge_kutta (a, eps, t, dt, param, cpdim, om, xx, yy, zz, x1, y1, z1);

```

```

t = 2.*dt;
for (j = 2; j < NSTEP; j++) {
    for (i = 0; i < NSYS; i++) {
        xx[i] = x1[i];
        yy[i] = y1[i];
        zz[i] = z1[i];
    }
runge_kutta (a, eps, t, dt, param, cpdim, om, xx, yy, zz, x2, y2, z2);

    for (i = 0; i < NSYS; i++) {
        if (x0[i] < x1[i] && x1[i] > x2[i]) nx[i] += 1.;
        if (y0[i] < y1[i] && y1[i] > y2[i]) ny[i] += 1.;
        x0[i] = x1[i];
        x1[i] = x2[i];
        y0[i] = y1[i];
        y1[i] = y2[i];
    }
    t += dt;
}

sumx = 0.;
sumy = 0.;
for (i = 0; i < NSYS; i++) {
    sumx += 2.*3.14159*nx[i]/(double)NSTEP/dt;
    sumy += 2.*3.14159*ny[i]/(double)NSTEP/dt;
}

mfq_x = sumx/(double)NSYS;
mfq_y = sumy/(double)NSYS;

sumxx = 0.;
sumyy = 0.;
for (i = 0; i < NSYS; i++) {
    sumxx += pow(2.*3.14159*nx[i]/(double)NSTEP/dt - mfq_x, 2);
    sumyy += pow(2.*3.14159*ny[i]/(double)NSTEP/dt - mfq_y, 2);
}

sd_frqx = sqrt(sumxx/((double)NSYS-1.));
sd_frqy = sqrt(sumyy/((double)NSYS-1.));

// fprintf(fout, " %6.4f %10.7f\n", eps, sd_frqx);
// fprintf(fout, " %8.4f %8.4f %8.4f\n", a, eps, sd_frqx);
// fprintf(fout, " %6.4f %10.7f\n", eps, sd_frqy);
// eps += depts;
// cout << eps << " " << sd_frqx << endl;
// cout << eps << " " << sd_frqy << endl;
}
a += da;
}
fclose(fout);
return 0;
}

```

A.9 Populations of Globally Coupled Nonidentical Oscillators with Variation of Topological Parameter

```

/* This program simulates phase synchronization in three-dimensional
globally coupled population of nonidentical Rossler oscillators
by computing the standard deviation of the observed frequencies
with variation of system parameter a.

Written by: Limin Qi, 03-25-2005
*/

#include <iostream.h>
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#define NPARAM 2
#define NSYS 500
#define NSTEP 1000
#define NEPS 31
#define NA 51

void rossler (double a, double eps, double t, double om[],
double param[], double cpdim[],
double sumx[], double sumy[], double sumz[],
double x[], double y[], double z[],
double dx[], double dy[], double dz[])
{
double b, c, cx, cy, cz;

b = param[0]; c = param[1];
cx = cpdim[0]; cy = cpdim[1]; cz = cpdim[2];

for (int i = 0; i < NSYS; i++) {
for (int j = 0; j < NSYS; j++) {
sumx[i] += x[j] - x[i];
sumy[i] += y[j] - y[i];
sumz[i] += z[j] - z[i];
}
dx[i] = -om[i]*y[i] - z[i] + cx*eps*sumx[i]/((double)NSYS - 1.);
dy[i] = om[i]*x[i] + a*y[i] + cy*eps*sumy[i]/((double)NSYS - 1.);
dz[i] = b + z[i]*(x[i] - c) + cz*eps*sumz[i]/((double)NSYS - 1.);
}
}

void runge_kutta (double a, double eps, double t, double dt,
double param[], double cpdim[], double om[],
double x[], double y[], double z[],
double xnew[], double ynew[], double znew[])
{
double f11[NSYS], f12[NSYS], f13[NSYS];
double f21[NSYS], f22[NSYS], f23[NSYS];
double f31[NSYS], f32[NSYS], f33[NSYS];
double f41[NSYS], f42[NSYS], f43[NSYS];
double xtem[NSYS], ytem[NSYS], ztem[NSYS];
double sumx[NSYS], sumy[NSYS], sumz[NSYS];
double t_half, half_dt, t_full;
int i;

for (i = 0; i < NSYS; i++) {
sumx[i] = 0.;
sumy[i] = 0.;
sumz[i] = 0.;
}

half_dt = 0.5 * dt;
rossler (a, eps, t, om, param, cpdim, sumx, sumy, sumz, x, y, z,
f11, f12, f13);

```

```

    t_half = t + half_dt;
    for (i = 0; i < NSYS; i++) {
        xtem[i] = x[i] + half_dt*f11[i];
        ytem[i] = y[i] + half_dt*f12[i];
        ztem[i] = z[i] + half_dt*f13[i];
    }
    rossler (a, eps, t_half, om, param, cpdim, sumx, sumy, sumz, xtem, ytem, ztem,
            f21, f22, f23);

    for (i = 0; i < NSYS; i++) {
        xtem[i] = x[i] + half_dt*f21[i];
        ytem[i] = y[i] + half_dt*f22[i];
        ztem[i] = z[i] + half_dt*f23[i];
    }
    rossler (a, eps, t_half, om, param, cpdim, sumx, sumy, sumz, xtem, ytem, ztem,
            f31, f32, f33);

    t_full = t + dt;
    for (i = 0; i < NSYS; i++) {
        xtem[i] = x[i] + dt*f31[i];
        ytem[i] = y[i] + dt*f31[i];
        ztem[i] = z[i] + dt*f31[i];
    }
    rossler (a, eps, t_full, om, param, cpdim, sumx, sumy, sumz, xtem, ytem, ztem,
            f41, f42, f43);

    for (i = 0; i < NSYS; i++) {
        xnew[i] = x[i]+dt/6.0*(f11[i]+f41[i]+2.0*(f21[i]+f31[i]));
        ynew[i] = y[i]+dt/6.0*(f12[i]+f42[i]+2.0*(f22[i]+f32[i]));
        znew[i] = z[i]+dt/6.0*(f13[i]+f43[i]+2.0*(f23[i]+f33[i]));
    }
}

int main()
{
    int    i, j, k, l;
    double xx[NSYS], yy[NSYS], zz[NSYS];
    double x0[NSYS], y0[NSYS], z0[NSYS];
    double x1[NSYS], y1[NSYS], z1[NSYS];
    double x2[NSYS], y2[NSYS], z2[NSYS];
    double param[NPARAM] = {0.4, 8.5}, om[NSYS];
    double cpdim[NCPDIM] = {1., 1., 1.};
    double a, da = 0.003, eps, deps = 0.005, t, dt = 0.1;
    double nx[NSYS], mfq_x, sumx, sumxx, sd_frqx;
    double ny[NSYS], mfq_y, sumy, sumyy, sd_frqy;
    unsigned seed;

    FILE *fout;
        fout = fopen("rNa.txt", "w");

    printf ("Enter a positive integer randomly: ");
    scanf("%u", &seed);
    srand(seed);

    for (i = 0; i < NSYS; i++) {
        om[i] = 0.9450 + 0.000025*(1 + (rand() % 5000));
        printf ( " om %8.4f\n", om[i]);
    }

    a = 0.05;
    for (l = 0; l < NA; l++) {
        eps = 0.0;
        for (k = 0; k < NEPS; k++) {

```

```

        t = 0.;
    for (i = 0; i < NSYS; i++) {
        nx[i] = 0.;
        ny[i] = 0.;
        x0[i] = -1.;
        y0[i] = 0.;
        z0[i] = 0.;
    }
    t = dt;
    for (i = 0; i < NSYS; i++) {
        xx[i] = x0[i];
        yy[i] = y0[i];
        zz[i] = z0[i];
    }

    runge_kutta (a, eps, t, dt, param, cpdim, om, xx, yy, zz, x1, y1, z1);

    t = 2.*dt;
    for (j = 2; j < NSTEP; j++) {

        for (i = 0; i < NSYS; i++) {
            xx[i] = x1[i];
            yy[i] = y1[i];
            zz[i] = z1[i];
        }

        runge_kutta (a, eps, t, dt, param, cpdim, om, xx, yy, zz, x2, y2, z2);

        for (i = 0; i < NSYS; i++) {
            if (x0[i] < x1[i] && x1[i] > x2[i]) nx[i] += 1.;
            if (y0[i] < y1[i] && y1[i] > y2[i]) ny[i] += 1.;
            x0[i] = x1[i];
            x1[i] = x2[i];
            y0[i] = y1[i];
            y1[i] = y2[i];
        }
        t += dt;
    }

    sumx = 0.;
    sumy = 0.;
    for (i = 0; i < NSYS; i++) {
        sumx += 2.*3.14159*nx[i]/(double)NSTEP/dt;
        sumy += 2.*3.14159*ny[i]/(double)NSTEP/dt;
    }

    mfg_x = sumx/(double)NSYS;
    mfg_y = sumy/(double)NSYS;

    sumxx = 0.;
    sumyy = 0.;
    for (i = 0; i < NSYS; i++) {
        sumxx += pow(2.*3.14159*nx[i]/(double)NSTEP/dt - mfg_x, 2);
        sumyy += pow(2.*3.14159*ny[i]/(double)NSTEP/dt - mfg_y, 2);
    }

    sd_frqx = sqrt(sumxx/((double)NSYS-1.));
    sd_frqy = sqrt(sumyy/((double)NSYS-1.));

    //
    fprintf(fout, " %10.7f\n", sd_frqx);
    fprintf(fout, " %8.4f %8.4f %8.4f\n", a, eps, sd_frqx);
    //
    fprintf(fout, " %6.4f %10.7f\n", eps, sd_frqx);
    //
    fprintf(fout, " %6.4f %10.7f\n", eps, sd_frqy);

```

```
        eps += deps;
//      cout << eps << " " << sd_frqx << endl;
//      cout << eps << " " << sd_frqy << endl;
    }
    a += da;
}
fclose(fout);
return 0;
}
```


REFERENCES

- [1] Rossler, O. E., “An Equation for Continuous Chaos”, *Physics Letters A*, Vol. 57, pp. 397-398 (1976).
- [2] Rosenblum, M. G., Pikovsky, A. S., and Kurths, J., “From Phase to Lag Synchronization in Coupled Chaotic Oscillators”, *Physical Review Letters*, Vol. 78, No. 22, pp.4193-4196 (1997).
- [3] Rosenblum, M. G., Pikovsky, A. S., and Kurths, J., “Phase Synchronization of Chaotic Oscillators”, *Physical Review Letters*, Vol. 76, No. 11, pp.1804-1807 (1996).
- [4] Sarasola, C., Torrealdea, F. J., D’Anjou, A., Moujahid, A., and Grana, M., “Energy Balance in Feedback Synchronization of Chaotic Systems”, *Physical Review E*, Vol. 69, No. 11, pp. 11606/1– 11606/12 (2004).
- [5] Zhan, M., Zheng, Z., Hu, G., Peng, X., “Nonlocal Chaotic Phase Synchronization”, *Physical Review E*, Vol. 62, No. 3, pp3552-3557 (2000).
- [6] Pastur, L., Boccaletti, S., and Ramazza, P. L., “Detecting Local Synchronization in Coupled Chaotic Systems”, *Physical Review E*, Vol. 69, No. 3, pp. 036201/1– 036201/4 (2004).
- [7] Osipov, G. V., Pikovsky, A. S., Rosenblum, M. G., and Kurths, J., “Phase Synchronization Effects in A Lattice of Nonidentical Rossler Oscillators”, *Physical Review E*, Vol. 55, No. 3, pp. 2353-2361 (1997).
- [8] Blasius, B., Montbrió, E., and Kurths, J., “Anomalous Phase Synchronization in Populations of Nonidentical Oscillators”, *Physical Review E*, Vol. 67, No. 3, pp.

035204/1–035204/4 (2003).

- [9] Li, C., and Chen, G., “Phase Synchronization in Small-World Networks of Chaotic Oscillators”, *Physica A*, Vol. 341, No. 1-4, pp. 73-79 (2004).
- [10] Sakaguchi, H., “Phase Transition in Globally Coupled Rossler Oscillators”, *Physical Review E*, Vol. 61, No. 6 B, pp. 7212-7214 (2000).
- [11] Zanette, D. H., and Mikhailov, A. S., “Dynamical Clustering in large Populations of Rossler Oscillators under the Action of Noise”, *Physical Review E*, Vol. 62, No. 6 A, pp. R7571-R7574 (2000).
- [12] Pikovsky, A., Rosenblum, M., Osipov, G., and Kurths, J., “Phase Synchronization of Chaotic Oscillators by External Driving”, *Physica D*, Vol. 104, pp. 219-238 (1997).
- [13] Kim, I., Kim, C-M, Kye, W-H, and Park, Y-J, “Phase Synchronization with Type-II Intermittency in Chaotic Oscillators”, *Physical Review E*, Vol. 62, No. 6 B, pp. 8826-8829, (2000).
- [14] Yang, H. L., “Phase Synchronization of Diffusively Coupled Rossler Oscillators with Funnel Attractors”, *Physical Review E*, Vol. 64, No. 2 II, pp. 26206/1-26206/5 (2001).
- [15] Yanchuk, S., Maistrenko, Y., and Mosekilde, E., “Loss of Synchronization in Coupled Rossler Systems”, *Physica D*, Vol. 154, No. 1-2, pp. 26-42 (2001).
- [16] Arai, K., Mizutani, S., and Yoshimura, K., “Deterministic Stochastic Resonance in A Rossler Oscillator”, *Physical Review E*, Vol. 69, No. 2 2, pp. 026203/1-026203/12 (2004).