

NLD - Non-Linear Dynamics and Chaos  
Physics 111B: Advanced Experimentation Laboratory  
University of California, Berkeley

## Contents

<b>1 Non-Linear Dynamics and Chaos Description (NLD)</b>	<b>1</b>
<b>2 Before the 1st Day of Lab</b>	<b>2</b>
<b>3 Reading Materials</b>	<b>3</b>
<b>4 Objectives</b>	<b>4</b>
<b>5 Introduction</b>	<b>5</b>
<b>6 Equipment used in this experiment</b>	<b>5</b>
<b>7 Theory</b>	<b>5</b>
7.1 The PN Junction: An Introductory Dynamical System . . . . .	5
7.2 Phase-Path Reconstruction . . . . .	8
7.3 PN-Junction phase plots & The Bifurcation Route to Chaos . . . . .	8
7.4 Fractals . . . . .	10
7.5 Flow and Liapunov Exponents . . . . .	11
7.6 Recurrence relations . . . . .	11
<b>8 The Experiment Apparatus</b>	<b>11</b>
<b>9 Procedures</b>	<b>12</b>
9.1 Getting started with LabView and Sampling . . . . .	12
<b>10 Graphing</b>	<b>14</b>
10.1 One-dimensional maps: Introductory chaotic systems. . . . .	15
10.2 Numerical calculation of Liapunov Exponents . . . . .	17
10.3 NLSIM.VI: the simulation VI . . . . .	18
10.4 Continuous-time systems- PN Junction . . . . .	19
<b>11 Bouncing Ball</b>	<b>23</b>
11.1 Information dimension . . . . .	25
<b>12 What your report should contain</b>	<b>26</b>
<b>13 The Custom VI's for this experiment</b>	<b>26</b>

## 1 Non-Linear Dynamics and Chaos Description (NLD)

Revised 2025-12-09

1. Note that there is NO eating or drinking in the 111-Lab anywhere, except in rooms 282 & 286 LeConte on the bench with the BLUE stripe around it. Thank You, the Staff.

This experiment is an introduction to non-linear dynamics, data acquisition, chaos theory and fractals. Limited as we are by our senses and relatively short powers of recall, much of the physical world seems aperiodic and defies quantitative description. While we have yet to discover closed form solutions to the simplest of systems (e.g. the one-dimensional gravitational three-body problem), the field of chaos reveals structure in their dynamics. The results of chaos theory have found practical applications in almost every branch of science.

In this experiment you will study the response of at least two different dynamical systems: a non-linear, damped harmonic oscillator and a system of op amps that reproduces the Lorentz attractor. You will measure their linear and non-linear behavior and write software to measure their information dimension.

You will learn the basics of digital sampling, Fourier transforms, geometric analysis of autonomous differential equations, information entropy, correlation dimension and the basics of programming in LabView (a data-acquisition and control language).

There are no prerequisites. But a strong background in differential equations and classical mechanics, coupled with a willingness to do a lot of reading will serve you well in this lab.

1. Pre-requisites: None
2. Days Allotted for the Experiment: 7

Pages of this lab.

1. **Non-Linear Dynamics and Chaos** (here)
2. **Experiment and Procedure**
3. **The custom VT's for this experiment & Appendix**

This lab will be graded 40% on theory, 20% on technique, and 40% on analysis. For more information, see the [Advanced Lab Syllabus](#).

Last updated 11/24/25 by Auden Young. Comments: E-mail [Dr Winthrop Williams](#)

## 2 Before the 1st Day of Lab

Complete the NLD Pre Lab found in the [Signature Sheet](#) for this lab. Print the signature sheet, discuss the experiment and pre-lab questions and answers with any faculty member or GSI, and receive their signature. In the course of the lab there will be examination points where you must STOP and get a GSI or professor to verify your understanding and/or verify proper experimental setup. You cannot skip these checkpoints, and must receive signatures demonstrating that you've consulted the staff. Some experiments may have mid lab questions that must be completed by specific days of the experiment. The completed [Signature Sheet](#) MUST be submitted as the first page of your lab report. Quick links to the checkpoint questions are found here: [1](#) [2](#) [3](#) [4](#) [5](#)

1. *Note: In order to view the private Youtube videos hosted by the university, you must be signed into your berkeley.edu Google account.*  
View the [Non-Linear Dynamics and Chaos Video](#).
2. Complete all of the mandatory prerequisite reading.

You should keep a laboratory notebook. The notebook should contain a detailed record of everything that was done and how/why it was done, as well as all of the data and analysis, also with plenty of how/why entries. This will aid you when you write your report.

- Last day of the experiment please fill out the [Experiment Evaluation](#)

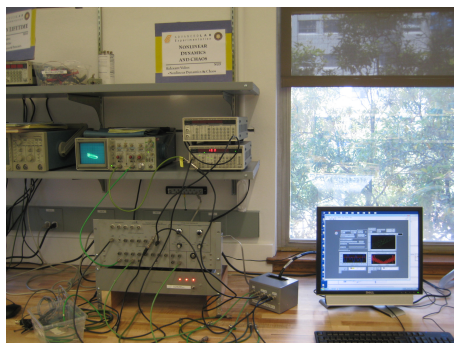


Figure 1: The NLD Chaos Experiment. See larger image [here](#)



Figure 2: DAQ box. See larger image [here](#)



Figure 3: The NLD apparatus wired for bouncing ball. See larger image [here](#)

### 3 Reading Materials

**Suggested Initial Reading:** Ref. [1]; Skim §1, §2.0-§2.2, §3.0-§3.4, §3.6, and then read §9.0, §9.2-9.4, §10.1-§10.5 (§10.6, §10.7 optional), §11, §12.0-§12.3

- [1] Strogatz, Steven H. *Nonlinear Dynamics and Chaos*. Addison-Wesley, New York. 1994. #Q172.5.C45 S767  
*This is an introductory book presuming no prior familiarity with the subject.*
- [2] Ott, E. *Chaos in Dynamical Systems*. Cambridge University Press. 1993. #Q172.5.C45O87  
*An excellent, broad, introductory book for advanced students.*
- [3] Schuster, H. G. “[Shannon’s Measure of Information](#).” From Appendix F in *Deterministic Chaos, an Introduction: 2nd ed.* VCH, New York. 1988. #QC174.84.S381  
*An introductory book appropriate for those with an advanced mathematical background.*
- [4] Press, W. T. et al. “[Fourier Transform Spectral Methods](#).” From Chapter 12 of *Numerical Recipes in C*. Oxford University Press, New York. 1988. #QA76.73C15 N865  
*A compendium of numerical algorithms for all branches of science. It is included here for its discussion of the Fourier transform algorithm.*
- [5] Zimmerman R. L. et al. “[The electronic bouncing ball](#),” Am. J. Phys. **60**, 370. April 1992.
- [6] Kadanoff, Leo P. “[Chaos: A View of Complexity in the Physical Sciences](#).” From *The Great Ideas Today*. Encyclopedia Britannica, Inc. 1986.  
*Excellent brief introduction.*
- [7] Hao, Bai-Lin. *Chaos: Chapter 1 - Chapter 6*. World Scientific Publishers. Singapore. 1985.  
*General introduction with reprints of papers in this field.*
- [8] Cvitanovic, P. “[Universality in Chaos](#).” Acta Physica Polonica, Vol. **A65**. 1984.
- [9] R. Van Buskirk and C. D. Jeffries, “[Observation of chaotic dynamics of coupled nonlinear oscillators](#).” Phys. Rev. A **31**, 3332. 1985.  
*A description of a driven on junction resonator, similar to that of the laboratory experiment.*
- [10] M. Henon, “[A Two-Dimensional Mapping with a Strange Attractor](#),” Communications in Mathematical Physics, Vol. **50**, 69-77 (1976).  
*Introduction of the Henon map; fractal structure.*
- [11] M. Feigenbaum, “[Universal Behavior in Nonlinear Systems](#),” Los Alamos Science, Vol. 1. 1980  
*Introduction to universality; the quadratic map.*

- [12] R. N. Bracewell, “[The Discrete Fourier Transform](#),” ch.18-The Fourier Transform and its Applications, 2nd edition, McGraw Hill (1978). *Brief Introduction to the fast Fourier Transform.*
- [13] J. Testa, J. Perez, C. Jeffries, “[Evidence for Universal Chaotic Behavior of a Driven Nonlinear Oscillator](#)”, Phys. Rev. Lett. **48**, 714-717 (1982). *Experimental results on the driven pn junction; essentially the experimental system used in the Physics 111-Lab Experiment.*
- [14] J. Theiler, “[Estimating fractal dimension](#),” J. Opt. Soc. Am. A **7**, 1055 (1990). *Another review article with an excellent bibliography. This one focuses on fractals.*
- [15] B. K. Clark et al., “[Fractal dimension of the strange attractor of the bouncing ball circuit](#),” Am. J. Phys. **63**, 157 (1995).
- [16] C. E. Shannon, [A Mathematical Theory of Communication](#), (University of Illinois Press, Urbana, 1964). #TK5101 .S45
- [17] R. May, “[Simple Mathematical Models with Very Complicated Dynamics](#),” Nature **261**, 459-467 (1976). *Important early review of nonlinear maps.*
- [18] J. Perez, and C. Jeffries, “[Effects of additive noise on a nonlinear oscillator exhibiting period doubling and chaotic behavior](#),” Phys. Rev. B **26**, 3460-3462 (1982). *Experiment results on adding noise to the driven pn junction.*
- [19] R. W. Hamming, “[Maximum Entropy](#),” The Art of Probability for Scientists and Engineers: Chapter 7, pp. 253-265 + 7 pages of diagrams
- [20] J. P. Eckmann, and D. Ruelle. “[Ergodic Theory of Chaos and Strange Attractors](#),” Rev. Mod. Phys. **57**, 617-656 (1985).
- [21] L. Wells and J. Travis *LabVIEW for Everyone*, Prentice Hall PTR (Available from Dr Winthrop Williams).
- [22] [Additional References and Resources](#)

## 4 Objectives

- Learn what real experimental physics is about
- Learn the synergy between experimental and theoretical work
- Learn to use pieces of equipment that are commonly used in research
- Learn how measurements are performed, analyzed, and interpreted.
- Learn how to present your work and results
- Learn problem solving strategies
- Learn how to manage and organize your time

**NOTE: You should be using LabVIEW 2025 to run all programs.**

## 5 Introduction

This experiment is an introduction to non-linear dynamics, data acquisition, chaos theory and fractals. Limited as we are by our senses and relatively short powers of recall, much of the physical world seems aperiodic and defies quantitative description. While we have yet to discover closed form solutions to the simplest of systems (e.g. the one-dimensional gravitational three-body problem), the field of chaos reveals structure in their dynamics. The results of chaos theory have found practical applications in almost every branch of science.

The term chaotic has come to mean something very specific in this field. It does not mean random or unorganized as the ordinary English word does. It refers instead to a particular type of dynamical behavior. One which at first sight appears random, but underlying it are ordered deterministic laws. Chaotic systems exhibit these key features: They are aperiodic, they exhibit sensitive dependence on initial conditions, and they are in some sense bounded.

## 6 Equipment used in this experiment

1. Function Signal Generator [SRS DS 345](#)
2. Computer interface Box
3. DAQ card in computer
4. Main Control Chassis NLD-86-35Rev2002
5. Bouncing Ball and Lorenz Attractor inside control chassis
6. Scope
7. Power Supply providing  $+15/-15/+5$  V
8. Rack for equipment

## 7 Theory

### 7.1 The PN Junction: An Introductory Dynamical System

A *dynamical system* is essentially anything that varies with time. The pertinent variables necessary to describe its motion are called the *dynamical variables* and the evolution of the system describes a path, or *trajectory* in the *state-space* (also called *phase space*) of the dynamical variables. Dynamical systems may be either *conservative*: a volume element in state space remains invariant (recall Liouville's theorem; see Marion, J. B. *Classical dynamics of particles & systems*); or the system may be *dissipative*: for which regions of state space become compressed as the systems evolves.

One of the dynamical systems we study is a driven damped oscillator with a non-linear response. Because it is damped this is a dissipative system. It consists of a pn-junction (a diode) connected in series with an inductor L, a resistor R, and a driving sine wave oscillator of voltage  $V_0(t) = V_{os} \cos(\omega_0 t)$ , see the left side of Figure 4.

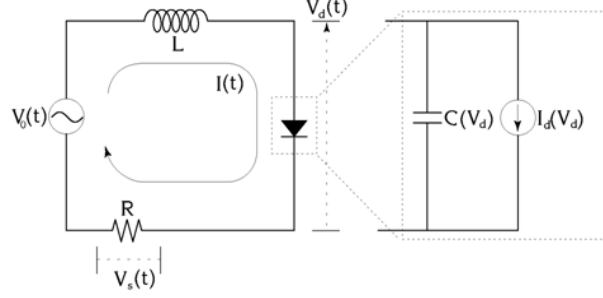


Figure 4: Driven damped oscillator circuit with a closeup of the diode.

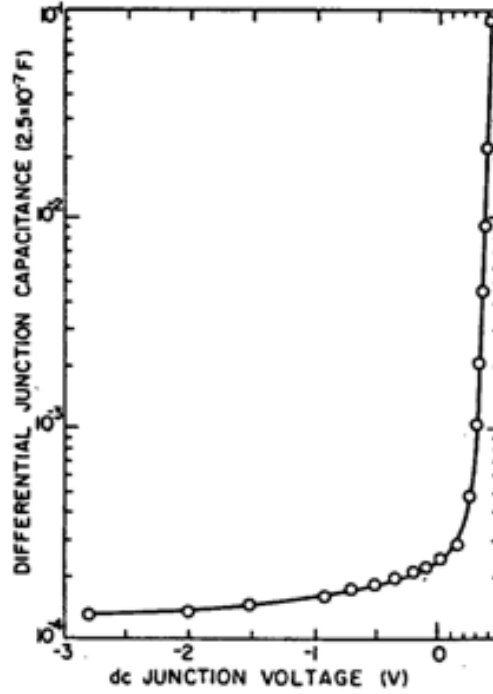


Figure 5: Capacitance of a typical diode as a function of diode voltage. From Ref. [7].

The diode is the non-linear circuit element: not only does it have an exponential  $I$ - $V$  characteristic,

$$I_d(V_d) = I_0[\exp(eV_d/kT) - 1] \quad (1)$$

but due to the properties of the junction (see Ref. [8]), it has a non-linear capacitance as well. The capacitance may be written as a function of the voltage across the diode,  $V_d$ :

$$C(V_d) = \begin{cases} C_0 e^{eV_d/kT} & : V_d > 0 \\ \frac{C_0}{\sqrt{1 - e^{eV_d/kT}}} & : V_d \leq 0 \end{cases} \quad (2)$$

Thus the diode may be modeled as a voltage-dependent capacitor connected in parallel with an ideal diode (a voltage-dependent current-source), giving the equivalent circuit on the right of Figure 4.

We use a computer to measure the voltage across the resistor  $R$  and the observed signal,  $V_s(t)$ , is proportional to the current  $I(t)$ .

For small values of driving voltage, the diode doesn't conduct, its capacitance is nearly constant, and the circuit behaves like a passive LRC resonator: A standard treatment of which yields a resonant frequency  $\omega_{\text{res}} \approx 1/\sqrt{LC(V_d = 0)} = 1/\sqrt{LC_0}$ . But as the driving amplitude increases, the increasing diode capacitance drops the resonant frequency significantly.

The effect of increasing driving-amplitude is seen in the following figures. They are plots of  $I(t)$  (measured as  $V_s(t)$  in Volts) vs. time (measured in sample number of a 50 kHz clock). In all three plots the driving oscillator is operating at the same frequency,  $\omega_{\text{res}} \approx 3.5$  kHz (a period of  $\approx 15$  samples).

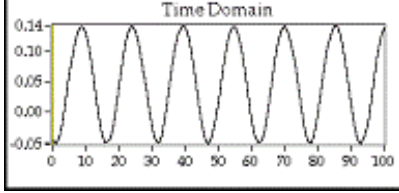


Figure 6:  $I(t)$ .  $V_{os} \approx 0.3$  V, response period  $\approx 15$  samples.

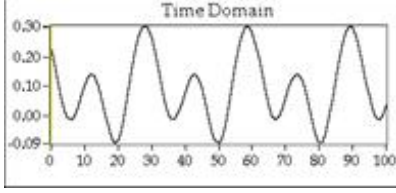


Figure 7:  $I(t)$ .  $V_{os} \approx 0.5$  V, response period  $\approx 30$  samples.

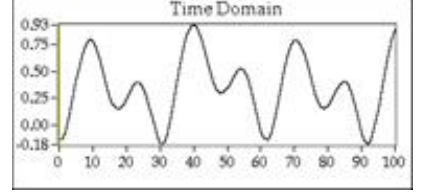


Figure 8:  $I(t)$ .  $V_{os} \approx 1.5$  V, response period  $\approx 60$  samples

In Figure 6,  $V_{os}$  is set to a small value and the response signal  $V_s(t)$  is a sine wave; this is the linear response. As  $V_{os}(t)$  is increased,  $V_s(t)$  begins to produce harmonic content at  $2\omega_0$ ,  $3\omega_0$  etc (not shown). But as you keep increasing the voltage, it will abruptly exhibit a change in behavior that, for reasons that will become clearer later, is called a *bifurcation*. This new state has *subharmonic* content  $\omega_0/2$  (Figure 7) and consequently is also called a *period-doubling* bifurcation. As  $V_{os}$  is further increased, this period doubling bifurcation occurs again and again at increasingly closely spaced values of  $V_{os}$ . As the period approaches infinity, the system becomes *chaotic*, i.e. it is aperiodic, with a small but visible noise-like component that looks like “jitter.”

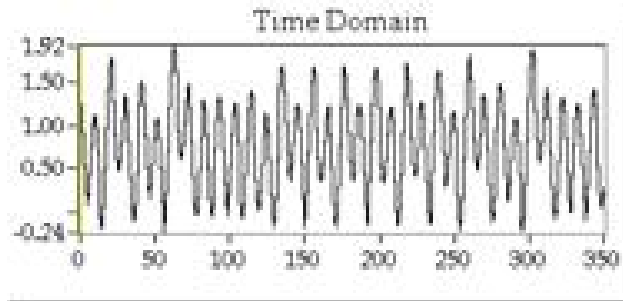


Figure 9:  $I(t)$ .  $V_{os} \approx 3.5$  V. The diode response current is now aperiodic.

As  $V_{os}$  is further increased, this “noise” dominates the trace; however, one finds narrow ranges of  $V_{os}$  (“windows”) for which this noise completely disappears, leaving a periodic signal. This noise is not random noise due to fluctuation phenomena (e.g., Johnson noise, shot noise, flicker noise), but is a natural consequence of the non-linear equations of motion of the system. As such, it is determined by the solution of the equations. We call this noise chaos. When the system becomes chaotic, Fourier spectral analysis shows a transition from a discrete to a continuous distribution.

The above behavior, a period-doubling cascade to chaos, is observed in many physical, chemical, and biological systems. To better understand it we shift our attention from the time domain to state-space.

The equations (of motion) for the PN-junction (Ref. [8], Sec. II) are:

$$\dot{I} = \frac{V_0 - RI - V_d}{L} \quad (3)$$

$$\dot{V}_d = \frac{I - I_d(V_d)}{C(V_d)} \quad (4)$$

$$\dot{\theta} = \omega \text{ (the time derivative of the driving signal's phase)} \quad (5)$$



Notice that these equations do not depend explicitly on time (and so are called *autonomous*). Instead they relate the rate of change of the current state to the current state itself. That is, given a state-space  $M$  of the variables  $(I, V_d, \theta)$ , we may write equations (3)-(5) as one vector equation

$$\dot{\vec{p}} = \vec{F}(\vec{p}) \text{ where } p \in M \quad (6)$$

Thus  $\vec{p}$  denotes the current state of the system, and  $\vec{p}(t)$  the path of the system through the three-dimensional state-space of the dynamical variables.

## 7.2 Phase-Path Reconstruction

If one could measure all the dynamical variables of a system simultaneously, the path  $\vec{p}(t)$  could be plotted easily. However, in many cases this is not possible and, moreover, it is not necessary in most cases. Look ahead to the reconstructed phase plots of Figures 10 to 13. These plots were made by sampling the diode current only. Rather than measuring the complete set of variables  $(I, V_d, \theta)$ , the trajectory is available in *each* variable (i.e. we could have sampled the diode voltage and come up with similar plots).

Sampling only one dynamical variable and reconstructing the trajectory in a higher-dimensional space is a process called *embedding*: Given a **discrete time series**  $\{I_n\}$  of regularly sampled values of  $I_d(t_0 + n\tau)$ , where  $t_0$  is an arbitrary initial time and  $\tau$  is the sample period, we embed the current in a higher dimension (appropriately called the *embedding-dimension*, in this example, dimension 3) by forming the points  $(I_n, I_{n+1}, I_{n+2})$  for all data.

This set of points then reveals an object topologically equivalent to the actual path in state-space. That this works is rather surprising. Ott presents a precise explanation (Ref. [2], section 3.8), but the basic idea is that if you know the current at a given moment,  $I_n$  that information gives you a little bit of information about where you are in state-space, that is, any point where the plane  $I = I_n$  intersects all possible trajectories. You then refine your knowledge by stipulating that you're at the point where not only is the current  $I_n$  but the current was  $I_{n-1}$  at a time  $\tau$  in the past. This pins down the location further, and subsequent restrictions on what the current was in the past refines your knowledge about what state it's in now.

This is by no means obvious, and it's complicated by the fact that there do exist situations where it doesn't work. But that the geometric properties of the system's trajectory can be found by sampling only one of the state-variables is remarkable and useful: It means that the structure of large-scale, multi-dimensional dynamical systems is available to a simple, single-detector apparatus.

## 7.3 PN-Junction phase plots & The Bifurcation Route to Chaos

Figure 10- Figure 13 show reconstructed phase-space trajectories,  $\vec{p}(t)$ , that correspond to the circuit's operation in Figure 6- Figure 9. The plots were made using the embedding technique described in the previous section, so the plot axes are  $(I_n, I_{n+1}, I_{n+2})$ . We speculate that  $\theta$  revolves around the line  $z = y = x$ , and that the other two state variables,  $V$  and  $I$ , correspond loosely to the distance along that line and the distance from that line. The graphs are all stereo-optic. Try holding them at arm's length (or set them on a desk as you stand over them) and letting your eyes focus at infinity. Let two of the double images overlap so that you can see three images. The middle one, if you get it to match up just right, will look like it's 3D.



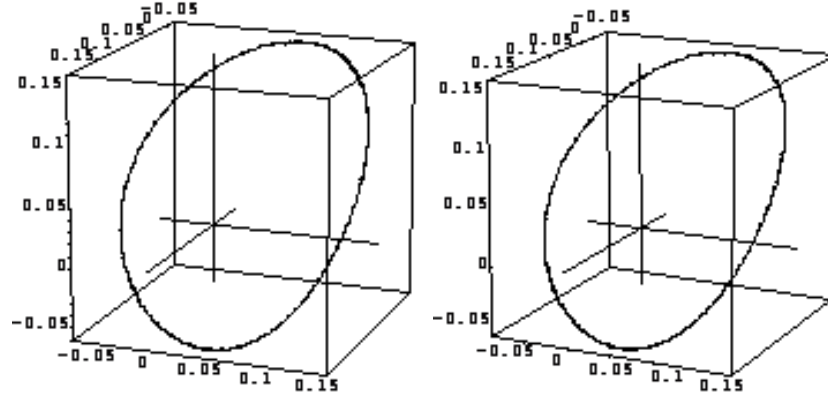


Figure 10: Left and Right stereo-optic images of the linear response of the PN-Junction. The path is almost an ellipse, which is the path of a simple harmonic oscillator.

For small values of the parameter  $V_{os}$ , the trajectory is a stable orbit. For larger value (after bifurcation to period 2) the orbit is still stable, but requires *two* cycles of the driving oscillator to return to its initial point (Figure 11). These trajectories are a subclass of a more general object called an ***attractor***: The subset of state space toward which a dynamical system evolves. Because they are periodic (the paths are closed cycles), the simple attractors of Figure 10-Figure 12 are known as ***limit-cycles***.

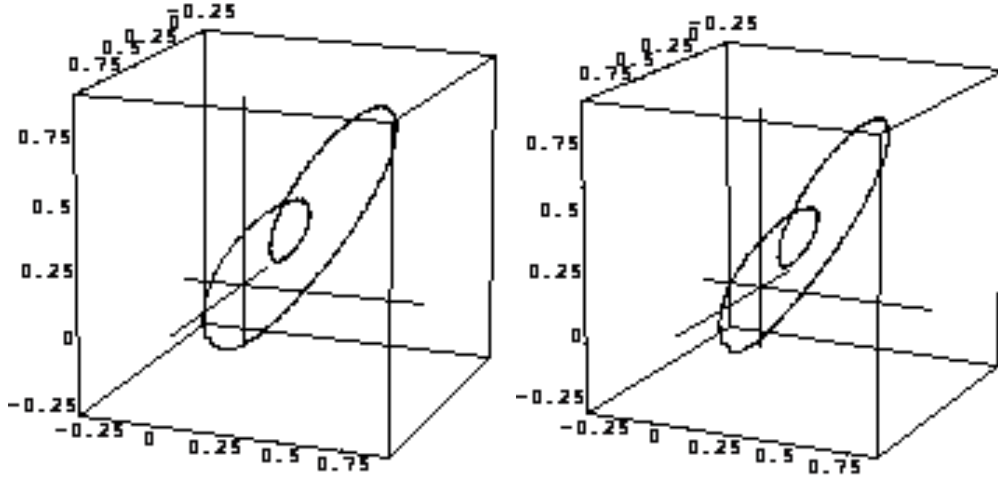


Figure 11: L & R images of the diode response after undergoing a period-doubling bifurcation

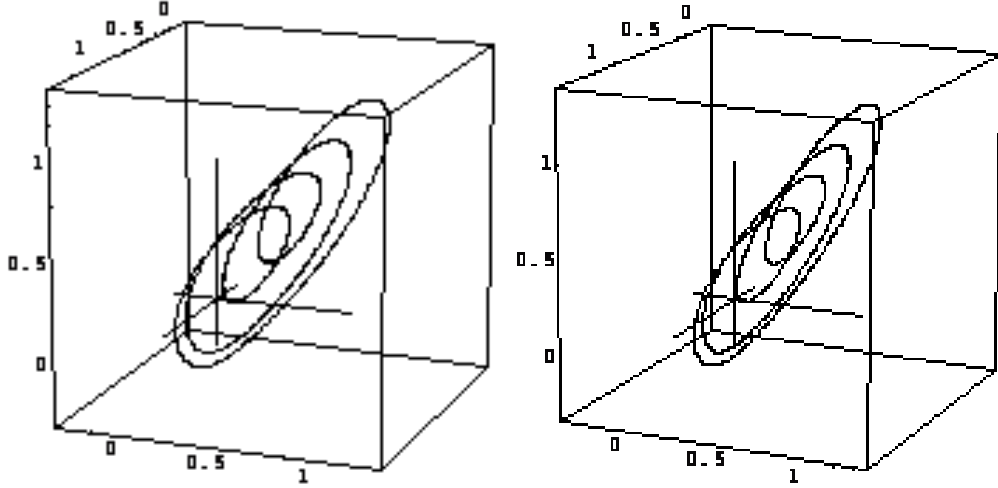


Figure 12: Same as above after another bifurcation.

Increasing  $V_{os}$  to the point where the system is chaotic reveals a more complicated attractor: it is a *fractal* object and hence called a *strange* attractor.

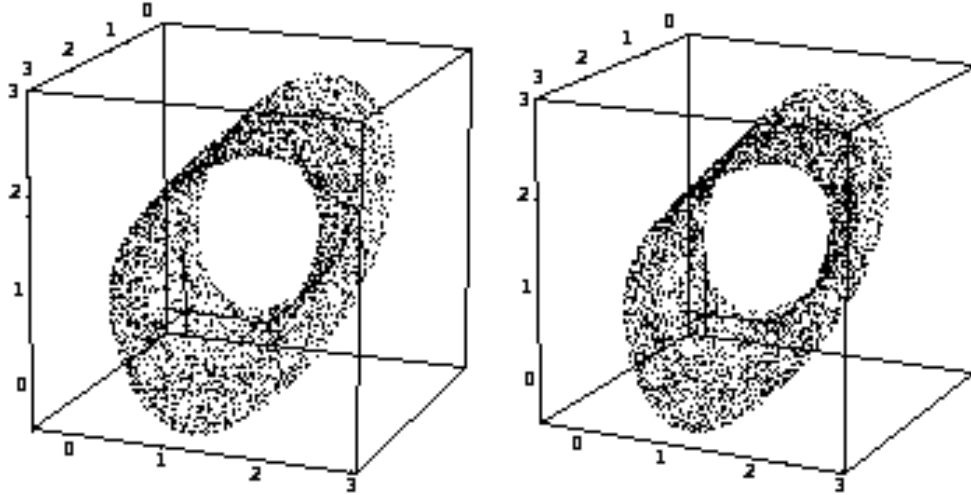


Figure 13: The system is now chaotic. The path in phase space never closes. And the evolution traces out the system's attractor. To improve the visibility of the plot the lines connecting the points have been omitted.

## 7.4 Fractals

As you learned in calculus, most geometrical objects become smoother as one gets closer and closer to them. Fractals, however, remain twisty and crumpled no matter how close one scrutinizes them. As such, they defy conventional definitions of length and volume. Efforts to quantify fractals first led to several different definitions of dimension and finally to generalized dimension (Ref. [3], Section 5.3, Ref. [4] p1060, Ref. [2], p78), the most intuitive of which is:

$$\text{bulk} \propto \text{size}^{\text{dimension}}, \text{ so that } \text{dimension} = \lim_{\text{size} \rightarrow 0} \frac{\ln \text{bulk}}{\ln \text{size}}$$

for appropriate definitions of bulk and size. Note that taking the limit as size  $\rightarrow 0$  means that this definition of dimension is a local property of the object and may vary over its extent.

For instance, in two dimensions a disk has an Area  $\propto \text{radius}^2$ , and  $\lim_{r \rightarrow 0} \frac{\ln \pi r^2}{\ln r} = 2$ . With this definition, objects of non-integral dimension are called fractals.

## 7.5 Flow and Liapunov Exponents

Chaotic systems are noted for several universal features: The bifurcation route to chaos; Stable windows of periodicity, the borders of which exhibit short bursts of chaotic signal (“intermittency”); And sensitivity to initial conditions. To help explain some of these features, we define a **flow**  $\phi_t$  that maps a state  $\vec{p}$  to its new state  $\phi_t(\vec{p})$  at a time  $t$  units later. This operation also satisfies the relationship:

$$\phi_t(\phi_s(\vec{p})) = \phi_{t+s}(\vec{p})$$

Now consider two neighboring initial conditions  $\vec{P}_0$  and  $\vec{P}_0 + \vec{h}$ . Expanding  $\phi_t(\vec{P}_0 + \vec{h})$  about  $\vec{P}_0$  gives  $\phi_t(\vec{P}_0 + \vec{h}) = \phi_t(\vec{P}_0) + \bar{J}(t)\vec{h} + O(\|\vec{h}\|^2) + \dots$  where  $\bar{J}(t)$  is the Jacobian matrix of partial derivatives of  $\phi_t$  with respect to  $\vec{P}_0$  and  $O(\|\vec{h}\|^2) + \dots$  are higher order terms. (In one dimension,  $\bar{J}(t)$  would just be  $\frac{\partial \phi}{\partial x}(t)$ ). Thus we see that a small displacement in initial conditions  $\vec{h}$  is magnified to  $\bar{J}(t)\vec{h}$ . The evolution of the distance between neighboring initial-conditions is then given by the eigenvalues of  $\bar{J}(t)$ . *Liapunov* (alternately spelled Lyapunov or Ljapunov) *exponents* are convenient measures of the eigenvalues and are defined as

$$\lambda_n \equiv \lim_{t \rightarrow \infty} \frac{\ln |n^{\text{th}} \text{ eigenvalue of } \bar{J}|}{t} \quad (7)$$

If the Liapunov exponents are all positive then the system diverges. If they are all negative then the system condenses to a point. If the largest one is positive and the sum of them all is negative, then it amounts to a stretch in one or more dimensions and a contraction in the others and chaotic behavior is guaranteed.

## 7.6 Recurrence relations

The equations of state (Eq’s (3), (4), (5)) are functions of a continuous time variable. Using the Runge-Kutta method of integrating ordinary differential equations, it is possible to form a set of discrete-time dynamical equations and these are seen to exhibit universal features common to other similar systems. For this reason, we study **iterative maps**: discrete versions of  $\phi_t$  that relate the next state of a system to its previous state. You might wonder why, if chaotic systems are so sensitive to initial conditions, should we expect to see any useful results computed by an approximating method such as Runge-Kutta. The reason is explained in [Ref. [1], §10.1], where the authors present the ‘shadowing’ lemma: Although a finitely computed chaotic time-series, beginning with an initial value, will rapidly diverge from the true orbit, there exists some other orbit that coincides closely to the computed series.

## 8 The Experiment Apparatus

This experiment uses an Intel CPU based computer, running the Windows operating system. The main tools are a data-acquisition computer-interface card and the LabView software package. The data-acquisition card has 8 input-channels, 2 output channels, and miscellaneous timing and TTL features. The inputs sample voltages in the  $\pm 5$  Volt range with 12-bit resolution at speeds to 100khz.

LabView is a graphical programming environment designed to facilitate lab-instrument control, data-acquisition, and rudimentary numerical analysis. There is a suite of LabView programs (called VIs; for Virtual Instruments) specifically written for this lab that constitutes the bulk of this experiment. You’ll also be writing a few of your own to get a better feel for LabView.

Most of the non-linear circuits you will examine in this lab are driven, so the computer is set up to modify a driving voltage, by using its DAC to attenuate a function-generator, and measure the circuit's response (see Figure 14). Because you won't be starting with the real systems, the details of the equipment and connections are deferred to section 4.5, where they should make more sense.

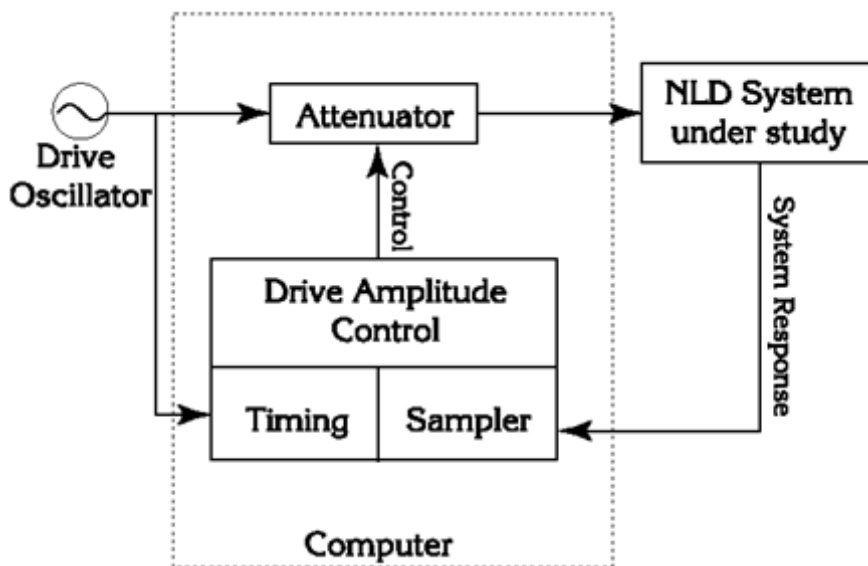


Figure 14: Simplified block diagram.

## 9 Procedures

### 9.1 Getting started with LabView and Sampling

Before starting LabView, create a directory for yourself inside My Documents folder. This is where you will save ALL your work. Read chapters 1-3 in LabView for Everyone and be sure to do the exercises. If you know LabView from BSC then proceed. Otherwise this will help immensely, especially if you have no previous programming experience. Here is a list of helpful things you might want to know:

- Control-H displays “pop-up” help, which will automatically explain things you point to with your cursor. Usually, this is sufficient, but if you need more detailed information, use the LabView manuals provided with the experiment.
- Control-E switches between the *front-panel* and the *diagram* windows.
- There are 3 palettes you will be using; *functions*, *controls*, and *tools*. Right clicking the mouse brings the appropriate palette for the window you're working in. *Functions* are used in the diagram window, *controls* are used in the front-panel window, and *tools* are used in both. Use the spacebar to switch between different tools in the *tools* palette.
- The LabView modules/elements (the things that look like icons) are unique in that there aren't two of them that look the same but do different things. Thus if your diagram looks just like the one in the lab manual, it will probably do the same thing.
- If you can't find an element/module, you may have to pick a similar one and then modify it.
- Wire from the input devices towards the output. Sometimes, the type of input to a module affects the output.

- Placing objects in the *front-panel* window places corresponding objects in the *diagram* window. It's a good idea to make the *front-panel* first, as then all of the corresponding objects are in place and you can plan what to do with them.
- Objects in the *front-panel* which represent numbers (i.e. slide bars, dials, etc.) or an array of numbers (i.e. charts, graphs) show up in the *diagram* window as [SGL] or [DBL] or [INT] (single or double precision, or integer).
- Modules in the *diagram* window have inputs and outputs, and it is important to wire them up correctly. The icon that looks like a spool of wire in the *tools* palette is used for wiring. Placing the mouse cursor over the different areas of the modules will tell you the specific functions of the inputs and outputs. These can sometimes be modified by right-clicking on them with the mouse.

Once you're ready to begin, copy all files to your my documents folder chaos. (Note: go to C:\support\NLD2013 for the NLD.llb file and copy all into your folder manually. Now start LabView (**Again, use LabVIEW 2016 for this part**) from the Windows Start menu. Your first assignment is to make a simple VI. Investigate sampling by putting a *waveform graph* and a *waveform chart* on the front panel. These are located under **Functions-F** which can be accessed by right clicking in an empty space in "Block Diagram".(in the same way the **Controls** can be accessed by right clicking in an empty space in "Front Panel".

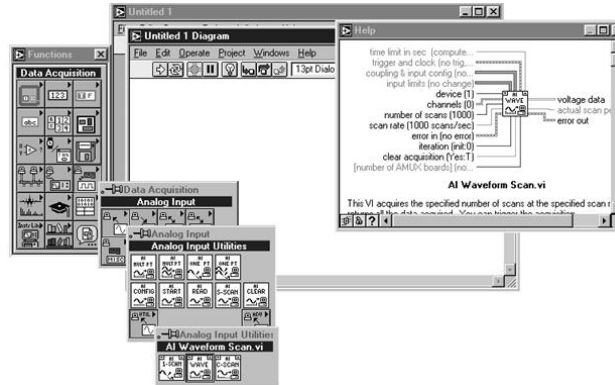


Figure 15: Pop-up help while inserting VI components.

**NOTE: AI Waveform Scan.vi is no more. Everything that requires it is optional. You should be able to find a similar VI in the new LabVIEW.**

Create a VI that reads a signal from the DS345 Function generator using the Computer DAQ box. Here's where to find the modules needed to create your program:

- Controls → Graph Indicators → Waveform Chart/Waveform Graph (you will need 2)
- Functions → All Functions → Measurement I/O → DAQmx → DAQ Assist
- Functions → All Functions → Numeric → Numeric Constant
- Functions → All Functions → Programming → Waveform → Analog Waveform → Power Spectrum
- Functions → All Functions → NI Measurements → Data Acquisition → Analog Input → Waveform Measurements → Spectral Measurements( you can search for this in Labview also)

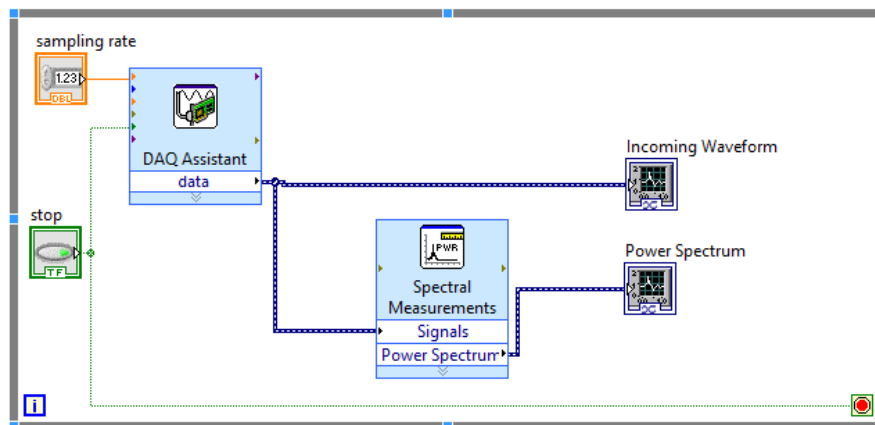


Figure 16: Connections for the simple sampling VI that you will make.

Once you have wired up the VI, you're ready to use it to sample signals. Use a function generator (Stanford Research Systems Model DS345) to send some signal to the Computer DAQ box. Connect the "function output" of the function generator to the terminal labeled AI 0 (make sure to terminate the signal). Adjust its output to produce an approximately 200 Hz sine wave, 5 Volts in amplitude (10V peak-to-peak). Use a scope to check the output. Once you create the DAQ Assistant it will prompt you for some inputs, select analog input and then voltage from the ai0 channel. The Spectral measurements one will also prompt you for some information (make sure you program it to spit out the power spectrum in a linear plot). Run your VI by pressing the "run" button.

Your two graphs should display the time-series of the signal (it should look like a sine-wave) and the *discrete* Fourier transform (DFT). We stress that it's the discrete transform because, while it is closely related to the continuous Fourier transform, there are some important differences that we hope you'll discover. The units of the time-series graph are Volts vs. Sample number. What are the units of the DFT window?

The Fourier transform of a sine wave is a delta-function (loosely speaking, a peak). Now change the frequency. Vary it between 100 and 10,000 Hz. See if you can explain what's happening. Also make sure that you are changing the sampling rate on your DAQ Assistant, think about why the sampling rate is important. Is there a way you can confidently predict a sampling rate that will give you an accurate reading of the signal? (Hint: Think back to 111A and the Nyquist criterion)

**Checkpoint DFT Window Units:** <sup>↑</sup> Call over a professor or GSI and show them your program and explain how it works. Explain the units on the DFT window, how do read the power spectrum from a time varying signal, and how you were able to accurately sample the incoming signal (explain what frequencies you set when varying the DS345).

## 10 Graphing

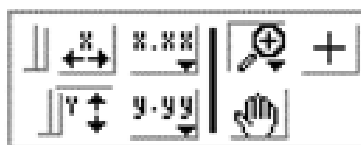


Figure 17: The graph-axes tool-palette.

As you experiment with the parameters of the exercise, you may want to change the range of the graphed data. For this purpose, each graph window has its own palette of display controls (Figure 17). These control

auto-ranging of the axes, format and precision of the axes, zooming, scrolling and the use of graph markers. See Ch 16 of the LabView User Manual; 2nd Ed.; Jeffrey Travis, for more information. You may also highlight (with the mouse) and type directly onto the axes scales.

Next, change the waveform from a sine to a triangle at 100 Hz. Calculate, or look-up in a reference, the Fourier series of a triangle wave. Does the displayed spectrum match what you expect? To help you determine the frequencies of different peaks, you should use graph cursors. You can enable them by right-clicking on the DFT window and selecting “show?cursor display.” This displays a palette of controls that permits the marking and measuring of values on the graph. See the LabView manual for details.

The harmonics of the triangle wave that occur above the Nyquist frequency, will be aliased. Adjust the frequency of the function generator between 100 and 800 Hz, and familiarize yourself with the operational details of aliasing.

## 10.1 One-dimensional maps: Introductory chaotic systems.

Multi-dimensional differential equations are hard to study. If they are insoluble, only numerical integration is possible. As simple examples of chaotic systems, one-dimensional maps offer an easy way to explore the defining features of chaos.

To familiarize yourself with 1-d maps, you will write a “cob-web” analyzer VI that shows the progress of iterations of the quadratic map (See Figure 18). Be sure to at least glance at §10.2 Ref. [1], to get an idea of what’s going on.

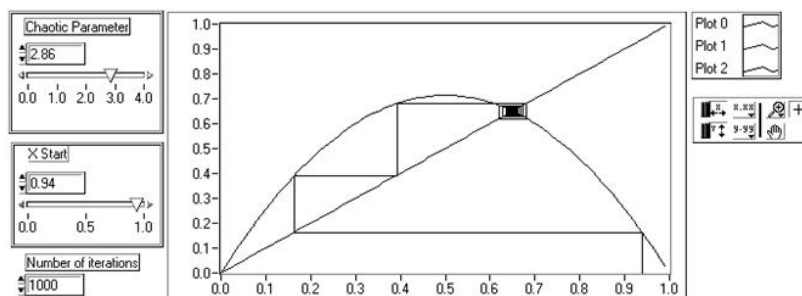


Figure 18: Cob web analyzer shows the effect of repeatedly applying the quadratic map to the initial value  $x = 0.94$ . It settles into a fixed point at about  $x = 0.65$

The graph displays three plots. The downward parabola is the quadratic map  $f(x) = rx(1-x)$ . The straight line is  $g(x) = x$ , and the stair-step or “cobweb” is the effect of successively applying the quadratic map to the initial point,  $x = 0.94$ . The recipe for producing the cobweb is as follows: Begin with the initial point  $x_0$ , follow a vertical line up to  $f(x_0)$ , follow a horizontal line over to the line  $y = x$ , you are now at the x-coordinate  $f(x_0)$ , proceed vertically to the graph of the quadratic map. You are now at the y-coordinate  $f(f(x_0))$ . Each time you move vertically to  $f$ , you have performed a further iteration of the map. Continuing in this way you can explore the long-term behavior of the map as you vary the initial condition and the chaotic parameter.

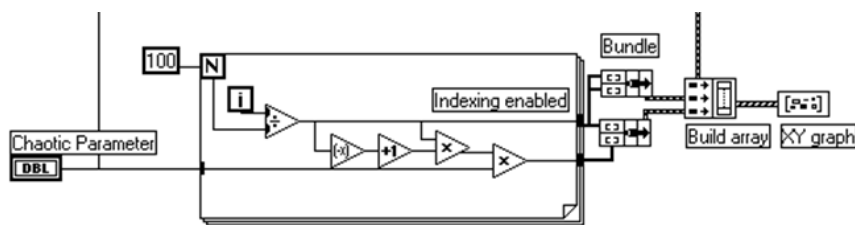


Figure 19: Section of the cob web VI



Making this VI yourself will teach you about arrays, or lists of numbers, in LabView. If you haven't programmed in a computer language before, you should know that each number in an array is indexed starting from 0 to  $N - 1$ . Here's the part of the cobweb VI that produces the graphs of  $f$  and  $g$  (Figure 19).

The big square is a for-loop that gets executed 100 times (the constant 100 wired to N). Each time through the loop the variable  $i$  increases by 1 ( $i$  starts at zero and goes through  $(N-1)$ ). The quotient is calculated and assembled into an array, which we shall call  $x_i$ , at the right border of the loop. The function  $f(x_i)$  is computed and also accumulated. The array is accumulated by means of the "indexing" feature at the border of the loop. Whenever you connect a wire from the inside to the outside of a loop a border node is formed to control how the data flows. Pop-up on the node to enable "indexing" and the result of each iteration will be assembled into an array. Leave indexing off and only the single number that was calculated last will be passed on.

Outside the loop, the arrays are "bundled" into a form that the XY-graph module will accept: an array of x-values bundled with an array of corresponding y-values. A "bundle" is a LabView data-structure that is a convenient way of collecting several things into one. It basically creates a list of ordered pairs,  $\{x_i, y_i\}$  which the program then builds into an array with three such lists. The graph reads these three lists of ordered pairs and plays connect the dots, making three graphs. One graph is made by taking the  $x_i$  array and bundling it with itself, producing the graph of  $y = x$ . The downward parabola,  $y = rx(1 - x)$ , is made by bundling  $x_i$  with  $f(x_i)$ .

That was the easy part, the graph of the cobweb part is a little harder and requires some discussion beforehand. We've already written a VI, named quad.vi, that calculates successive iterates of the quad map. The VI is located in C:\NLD-chaos\LABVIEW\CHAOS. So let's assume we have a series of map-iterates

$$m_i = \{m_0, m_1, m_2, \dots, m_n\} = \{x_i, f(x_i), f(f(x_i)), \dots, f^n(x_i)\}$$

We arrange these values in a way that will let LabView draw the cobweb by connecting the points:

$$\{x_i, y_i\} = \{(m_0, 0), (m_0, m_1), (m_1, m_1), (m_1, m_2), (m_2, m_2), (m_2, m_3), \dots, (m_{n-1}, m_{n-1}), (m_{n-1}, m_n)\}$$

The  $x$ -values form a series  $\{m_0, m_0, m_1, m_1, m_2, m_2, \dots, m_{n-1}, m_{n-1}\}$  which we may form by "interleaving" the  $x_i$  array with itself and removing the last two elements. Fortunately, LabView has a function that interleaves arrays and a function that takes an array's subset.

The  $y$ -values of the cobweb form a series  $\{0, m_1, m_1, m_2, m_2, \dots, m_{n-1}, m_n\}$  which can also be made by interleaving the  $x_i$  array and taking subsets (and replacing one element with a zero). Once you have the two arrays of  $x$ 's and  $y$ 's, you must bundle them together and then build an array out of this bundle and the other bundled plots of  $f$  and  $g$  (see the right side of Figure 19).

Try making the cobweb analyzer yourself. First, understand and make the section in Figure 19. Make the "Build array" icon have only two inputs, and then run the program. It should give you a graph of  $y = x$  and  $y = rx(1-x)$ .

Next, expand the "Build array" icon to three inputs and start the cobweb part. Be sure to look at quad.vi to see how it works. You can add a VI into your diagram by going to the *functions* palette and clicking "Selecting a VI...". You can then treat it as any other module with inputs and outputs. Consequently, all of the inputs in quad.vi will need to show up on your *front-panel* except for the *pre-iterate* input which you should just wire with a zero in your diagram. There are lots of different ways to program the cobweb, but you'll probably find the array functions "Build Array," "Interleave 1-D Arrays," "Replace Array Elements," "Array Size," and "Array Subset" helpful. Read what these modules do, and then make a block diagram on paper illustrating how you would use these functions to make the necessary  $x$  and  $y$  arrays for the cobweb. Then wire it up in LabView, bundle them together as before, and input it into the "Build Array" next to the other two bundles. Be sure to pay attention to whether your inputs/outputs are single elements or arrays of many elements. Some of the module inputs need to be told whether to accept elements or arrays.

Again, try to do this on your own. If you get really stuck, you can look in appendix A of this write-up for hints. Once you have the working VI, you can use it for a variety of things. Observe how the derivative of  $f(x_i)$  affects the stability of fixed points (Ref. [1] §10.1). Look for limit cycles, period windows, etc. Change the function  $f(x_i)$  to  $f(f(x_i))$  or some other 1-d map and observe the same qualitative behavior. To do this, you will have to modify both your program and quad.vi. Save quad.vi as quad2.vi in your directory rather than messing with the original. Try any non-invertible function that takes the unit interval to itself. There are numerous ones in the references: the tent map, the shift map, the sine map, etc. Try to come up with one of your own. Pretty much any function with a range of the unit interval that starts at (0,0) and goes to (1,0) will work. The maps must be non-invertible because there must be some dissipation in the equations [Ott [2] p8]. Try at least one other map and analyze it.

**Checkpoint** Cob Web Program:<sup>↑</sup> Call over a GSI and Professor and show them your program. Explain what you have done.

## 10.2 Numerical calculation of Liapunov Exponents

Sensitive dependence on initial conditions is one of the defining features of chaotic behavior. Consider two close initial-conditions,  $\xi$  and  $\xi + \delta$  evolving in time. If their distance of separation increases exponentially, i.e.  $|f(t, \xi) - f(t, \xi + \delta)| = \delta e^{\lambda t}$ , we say the function  $f$  exhibits sensitive dependence on initial conditions, and this case, the coefficient is necessarily positive. More generally the exponent is called the Liapunov exponent (see Ref. [1], section 9.3 especially the qualifications after fig 9.3.5). As such, if you can calculate or measure the exponent, you have a test to determine whether or not a particular system is chaotic. If is not positive, the system is not chaotic.

As previously mentioned, the Liapunov exponents of multi-dimensional systems are the average eigenvalues of the Jacobian matrix of the system (see also Theiler [14] eq's 4-7). So rather than solving an eigenvalue problem on a multi-dimensional system, it's easier to start with a one-dimensional map. The Liapunov exponent for a 1-d map is given by (Ref. [1], p367). Following example 10.5.3 in Ref. [1], write a python script to calculate the Liapunov exponent of the quadratic map as a function of its chaotic parameter. You will probably find it useful to know that the inputs to LabView's arithmetic operators are *polymorphic*. For example you can wire an array and a constant to the "add" function and the result is a new array whose elements are the old array with the constant added to each of them. Most of the math functions work this way for instance the log function can take an array of numbers and produce an array of logarithms of the numbers (see Figure 20). You will make use of quad.vi again.

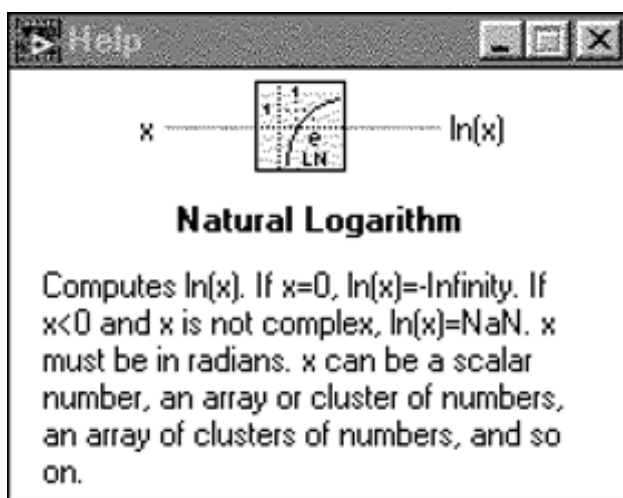


Figure 20: LabView functions can take arrays as inputs, producing arrays of results.

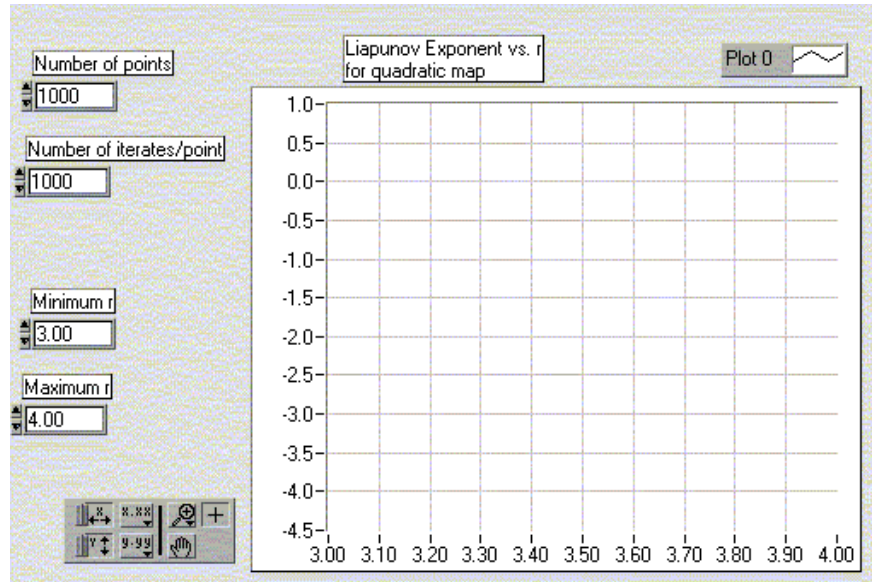


Figure 21: Liapunov exponent VI.

The front panel of your finished VI should look something like Figure 21. After you look at the algorithm in Ref. [1], make another block diagram of what you'll want to do and then try it in LabView. You'll probably find a FOR loop helpful. If you find yourself spending more than an hour or so on it, you should look at the solution in the appendix. When you have your working VI, explore the area around  $r = 3.57$  especially. What happens when  $\lambda \rightarrow 0$  Use your observations to approximate Feigenbaum's ratio  $\delta$ . Ref. [1] suggests 10,000 iterations, but this is a bit excessive, not to mention time consuming. Make sure to have lots of different values for  $r$ , the chaotic parameter. Experiment to see what gives you good graphs.

**Checkpoint Liapunov Exponent Program:** <sup>↑</sup> Call over a GSI and show them your program. Explain what you have done and which value of the chaotic parameter seems to work best.

### 10.3 NLSIM.VI: the simulation VI

Before proceeding to real-time NLD systems, you will take some time to run the simulation VI, NLSIM.VI. Open NLSIM.VI. This VI (Figure 22) allows you to display the time-series, return-map and power spectrum of a variety of signals. As in all the VI's, press control-H and move the mouse pointer over the elements of the screen to get specific information on how to use the program.

Start the VI, select the cosine function, and focus your attention to the return map. Many dynamical systems are described by a cosine function, for example the simple harmonic oscillator of a mass on a spring:  $x(t) = A \cos(\sqrt{\frac{k}{m}}t)$ . What are the axes of the phase space of this system? What is the shape of the paths in phase space for this system? Is it similar to what you see in the return map? What effect does changing the sample rate have on the shape of the trajectory? When you increase the sample rate, why does the return map narrow around the line  $y = x$ ? What happens if the sample rate is a multiple of the frequency of the cosine? Now try the function  $\cos(\omega_1 t) + \cos(\omega_2 t) + a \cos(\omega_1 t) \cos(\omega_2 t)$ . The return map for this function is more complicated: When  $a$  is zero, it's a two-dimensional uncoupled harmonic oscillator and its trajectories are confined to a torus in phase space (c.f. Ref. [1], §8.6). What conditions are necessary for the return-map to show a projection of a torus and not a cycloid? Be sure to explain what you're seeing in the Time Domain and Frequency Domain. Is it chaotic?

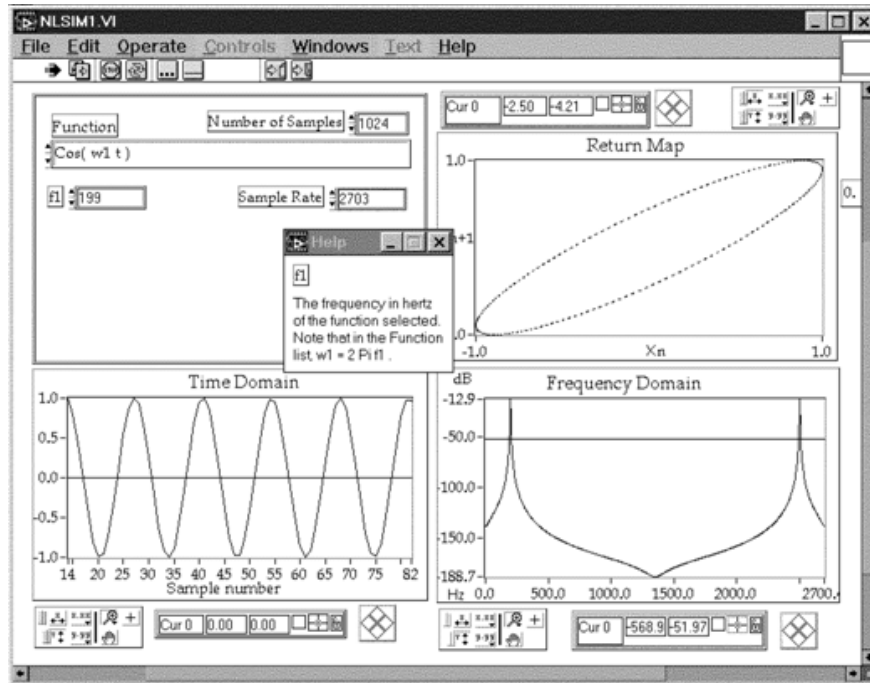


Figure 22: The simulation VI.

Now select the Quadratic Map function and set the parameter  $r = 2.8$ . The return map should show a single point. Increase  $r$  to about 3, the map will bifurcate, and you should see two points on the return map. Continuing to increase  $r$  will produce successive bifurcations and finally chaos at  $r = 3.5$ . One dimensional maps like this one are good analogs to *Poincaré sections* and when you sample the real NLD systems synchronous to their driving forces, you will see the same behavior. So it pays to spend some time here to really figure out what's going on. Be sure to compare this map with your cobweb VI at the same value of  $r$ . Also look at the quadratic map's bifurcation program.

**Checkpoint** Henon Map:<sup>†</sup> Take a look at the Henon Map. Play with the non-linear parameter between 0 and about 1.4. What is that funky thing in the Return Map? Look at the Henon map's bifurcation program as well. Discuss what you are thinking with a GSI or Professor.

## 10.4 Continuous-time systems- PN Junction

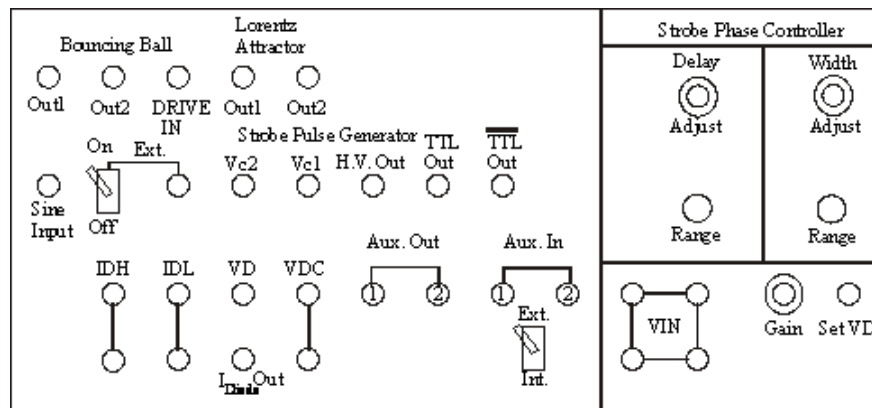


Figure 23: The Physics 111-Lab Nonlinear Dynamics Laboratory Panel.

Now we're ready to proceed to real-time continuous systems. At present there are two driven NLD systems at the experiment station: The bouncing ball circuit and the PN-junction. For both these driven systems, the basic set-up is the same: The computer controls the driving amplitude and measures the system's response. Both circuits, as well as most of the supporting electronics are located behind the large panel marked "Nonlinear Dynamics Laboratory." (see Figure 23) Note: the Sine Input switch should be off for the PN-junction part.

The drive oscillator is an Stanford Research System Model DS345. It is fed into the external reference pin  $V_{\text{ref}}$  of the DAC on the computer. Recall from your basic electronics experience that when you set a DAC to a specific level, e.g.  $j$  out of a maximum of  $N$ , the DAC outputs a voltage equal to  $V_{\text{DAC}} = V_{\text{ref}} \frac{j}{N}$ . Thus if  $V_{\text{ref}}$  is a sine-wave varying in time, the DAC outputs a sine wave of attenuated amplitude. The computer modified sine-wave is then fed to the NLD system under study. The NLD system's response is sent to the ADC input of the computer where it can be sampled either synchronously or asynchronously with respect to the driving frequency. When sampling is performed asynchronously, an internal timebase (not shown) is used. Synchronous sampling is achieved by a "strober," which sends TTL pulses to the EXTERNAL TRIGGER pin of the ADC whenever the drive signal achieves a certain phase (adjustable as "Pulse delay" on the NLD box) in its cycle. The sine-wave is also connected (via the square-wave converter on the NLD box) to the computer's timer pin (labeled GATE0) for computer measurement of the drive frequency.

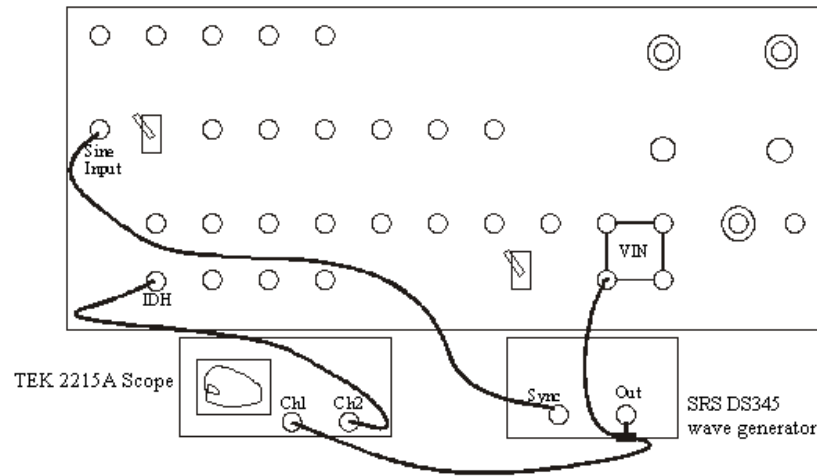


Figure 24: Connecting the scope to see the phase portrait of the PN-junction.

The PN junction circuit requires power, so turn on the system with the switch on the power supply located at the top of the rack. Note: You should hook up the TTL bar output to the Z-axis input on the back of the scope. You will also need to have the Sine Input Switch Off.

Let's begin with the PN Junction: Make the following connections referring, in addition to Figure 24:

- $\overline{\text{TTL}}$  to the back of the Tektronix Dual Trace 2215A scope
- Set the SRS DS345 to  $f = 3500$  Hz and 10 volts, VPP output, Terminate with  $50\Omega$ .
- Set the STROBE PHASE CONTROLLER "DELAY RANGE" to  $30 \mu\text{s}$  and "ADJ" to 0.0 (fully CCW)
- Set the STROBE PHASE CONTROLLER "WIDTH RANGE" to  $3 \mu\text{s}$  and "ADJ" to 0.0 (fully CCW)
- Set PN JUNCTION DRIVER "GAIN" to 0.0 (fully CCW, counter-clockwise).

You should now see a horizontal line on the scope. Increase the PN JUNCTION DRIVER GAIN by turning the dial clockwise and the scope should display an ellipse. (If you are having trouble getting it to display an ellipse - i.e. after you have checked your cable connections and scope settings **thoroughly** - for this section

only you may proceed to the DAQ portion. If it still does not work after being connected to the computer instead of the scope, you have done something wrong/call a GSI for help.) Having connected  $V_o(t)$  and  $V_s(t)$  (recall the notation of the introduction) to the  $x$ ,  $y$  axes of the oscilloscope (remember to put the time dial on x.y). The displayed graph is a plot of  $I$  vs.  $V_{os} \cos(\theta)$ , a two-dimensional projection of the system's path in phase space.

At such a low driving amplitude, the diode is responding linearly: The diode current is periodic at the same frequency as the drive oscillator with a small phase offset. Observe the behavior as you increase the DRIVER GAIN- it should follow the discussion of the pn-junction earlier in this write-up and the video.

The scope highlights sections of the plot according to the STROBE PHASE CONTROLLER adjustments. Hook up the TTL bar output to the Z-axis input on the back of the scope. Make sure the scope intensity is not too bright and you should be able to see the dots corresponding to points of constant phase of the drive. Recall that the state-space for this system is  $(I, V_d, \theta)$  so a set of points sampled at a constant  $\theta$  constitute a **Poincaré section**, a subset of state space that slices the attractor non-tangential to the trajectories. The Poincaré section will contain as many dots as there are loops. Changing the DELAY potentiometer moves the slice along the attractor. When you sample asynchronously, the computer will sample coincident with the dots.

Now let's include the computer. All of the connections to the computer are made through a single box, called the *Computer Interface Box* (Figure 25).

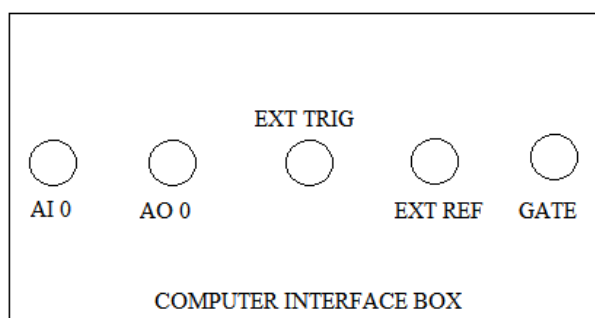


Figure 25: The connections on the computer interface box.



- SRS DS345 “50Ω; Sine output” —————to “EXT REF”
- COMPUTER “AO 0”—————to PN JUNCTION “ $V_{in}$ ”
- PN JUNCTION “Differential  $I_{diode}$  out”—————to “AI 0”
- STROBE PULSE GENERATOR “TTL OUT”—————to “GATE”
- STROBE PULSE GENERATOR “TTL OUT”—————to “EXT TRIG”
- Sine Input switch set to —————Off
- set the switch below AUX in/out —————to “INT”

This VI controls the drive amplitude, collects a sample of data points, and presents three different views of the response: a time-series, Fourier power spectrum, and return-map. We suggest you use 30kHz for the sampling rate. The more points you take in your sample, the better your DFT resolution will be, but the longer the computer will take to calculate and display. So to start, select 256 points. **You should make sure that the product of “Number of points per value of V” and “#” of points” does not exceed  $2^{31} - 1$ .** Vary the drive amplitude and estimate Feigenbaum’s ratio based on the system’s bifurcation points. Compare the return map to the picture you see on the scope. There are two other features of this VI: You can save the data to a file, and send the data to a subroutine that calculates information dimension. We’ll discuss these options later.

22



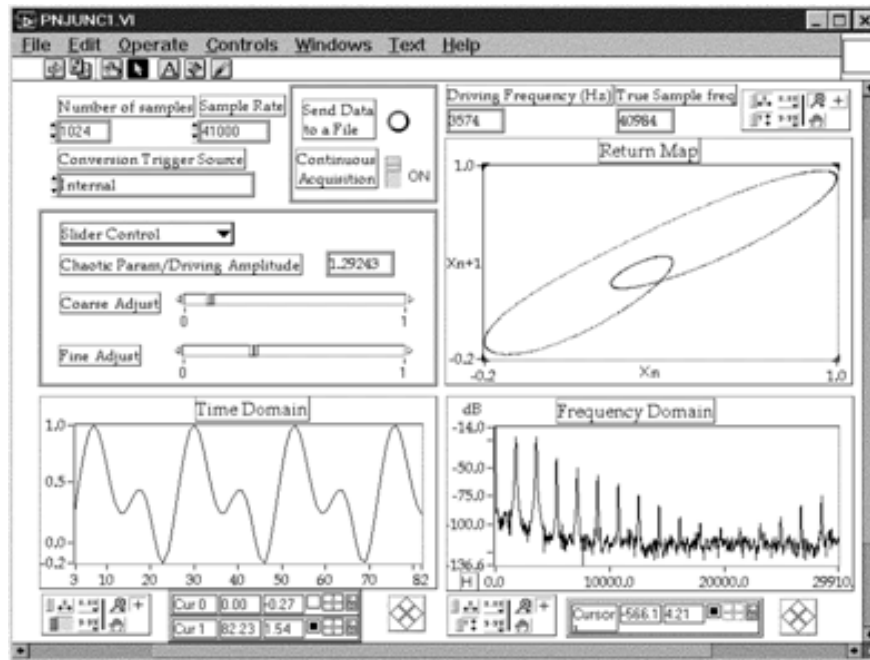


Figure 27: NLDGEN.VI (similar to pnjunc1.vi)- For sampling real-time driven NLD systems.

## 11 Bouncing Ball

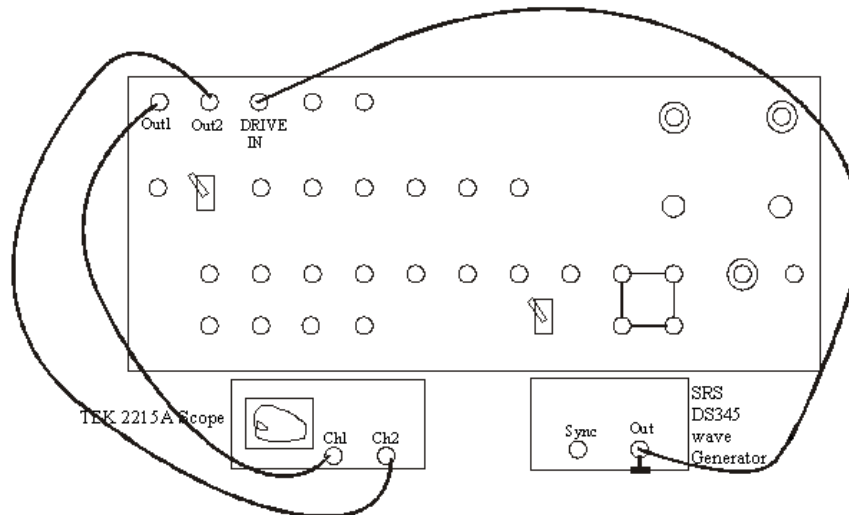


Figure 28: Connections for Bouncing Ball Circuit.

Let's now turn our attention to the bouncing ball circuit (BB).

- Combine a terminator on a T and connect the SRS DS345 with a sine wave out to the DRIVE input on the panel.
- Connect BB OUT1 to the Tek 2215 scope CH 1.
- Connect BB OUT2 to the Tek 2215 scope CH 2.

Now we're ready to hook up the computer. Make/check the following connections, referring to Figure 29:

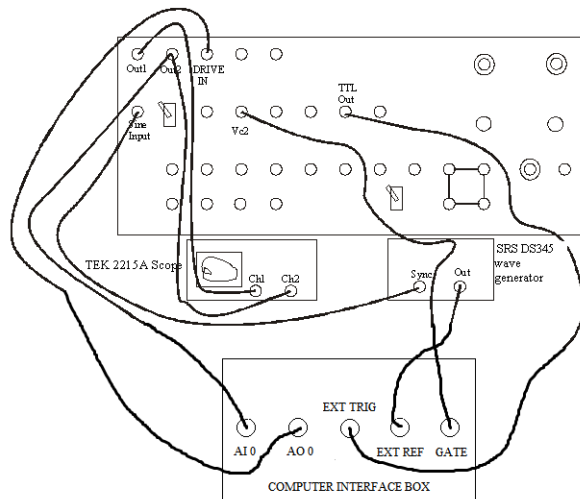
[illegible]

Figure 30: Front panel of Nldgen.vi in LabVIEW 2015 for Bouncing Ball.

- Disconnect the PN-JUNCTION setup

24

- DS345 “FUNCTION” output————— to “EXT REF”
- DS345 “SYNC” output————— to STROBE PULSE GENERATOR “SINE INPUT”
- “AO 0”————— to Bouncing ball “DRIVE”
- Bouncing ball OUT 1————— to scope CH1
- Bouncing ball OUT 2————— to scope CH2 and “AI 0”
- STROBE PULSE GENERATOR “VC2”—— to “GATE”
- STROBE PULSE GENERATOR “TTL OUT”- to “EXT TRIG”
- Set the Sine Input Switch to—————Off
- Set the DS345 to  $f = 90\text{Hz}$  and full amplitude output.
- Run the NLDGEN VI and start by sampling 256 points at 500 Hz

**Checkpoint Bouncing Ball and PN-Junction Circuits:**<sup>†</sup> What is fundamentally different about the chaotic behavior of the bouncing ball circuit when compared to the PN-Junction? (Hint: What would the bifurcation diagram for this circuit look like? Does it even have one?) Discuss what you notice with a Professor or GSI.

## 11.1 Information dimension

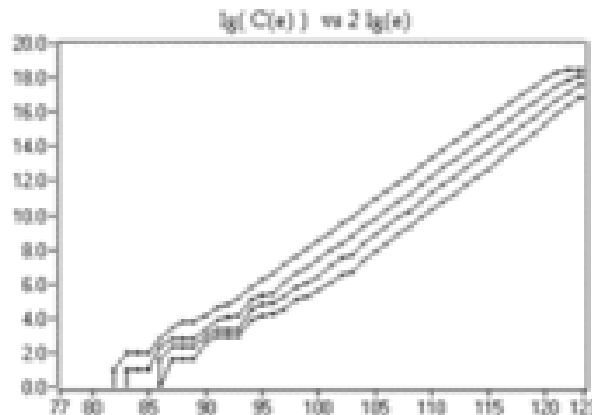


Figure 31: Data from the Quadratic map, analyzed by the information-dimension VI.

Chaotic systems have fractal attractors and a commonly measured quantity is the attractor’s information (or correlation) dimension (Ref. [1], §11.5, Ref. [14], pp1062-1063, pp1066-1070). The correlation dimension measures how the state-space “mass” of the attractor scales with length. The program `python_calccorr.vi` computes  $C(N, \epsilon)$ , the number of points within a distance  $\epsilon$  of each other, for increasing embedding dimensions and then displays  $\log_2 C(N, \epsilon)$  vs.  $\log_2 \epsilon$  (Figure 31). The slope of the linear region of the plot is then the information dimension. See the article by Theiler [14] for full details and Ref. [9] for a simple application of these concepts. In the Theiler article, read through the section on the Grassberger algorithm to understand the parameters for the program.

- As an introductory exercise, do the following: run `python_calccorr.vi` with “Random number generator” selected. Begin with an epsilon resolution of 0.01, an epsilon max of 0.125, a max embedding dimension (m) of 2, points (N) set to 2048, elapsed time (tau) set to 1, and near neighbors to skip set to 1. (If you want to save the result, select save to file.) Run the program. Try playing around with the parameters above based on your understanding from the Theiler article. What happens as m increases? As epsilon resolution decreases?

- When you have an understanding of the parameters using the random number generation, run NLSIM.vi with functions other than the random number generator. Send the data to a file, and then in python\_calccorr.vi de-select random number generator. When you run the vi, it will give you the option to import a file - import the file you saved from NLSIM.VI. What do you see? Use the graph cursors to estimate the slope of each line. How does each slope compare to the line's embedding-dimension?

The data calculated by python\_calccorr.vi can be saved to a data file for more accurate linear fitting (it will save a file for each value of  $m$  between 1 and the max embedding dimension that you set). Simply press the "save to file" button, name the file, and open it from within Excel (or copy it to a disk and take it to your favorite software). You should measure the information dimension of the PN-junction and the Bouncing Ball circuit for varying driving amplitudes and frequencies. Are the values what you expect?

## 12 What your report should contain

Your report should be an introduction to NLD and Chaos and a presentation of experimental results. Most of this material about chaos is unfamiliar to the majority of undergraduates. Thus your goal should be to illustrate the concepts set forth in the reading by relating them to what you did in lab.

The results you should present are *at least* the following: Experimental confirmation of the bifurcation route to chaos. The value of Feigenbaum's scaling constant  $\alpha$  and  $\delta$  measured from a variety of sources. What does Feigenbaum's ratio mean? The information (Lyapunov) dimension of the attractors of the PN-junction and the Bouncing Ball circuit as compared to that of conservative (Hamiltonian) systems. How do the Fourier Transform, the frequency spectrum, and the power spectrum differ?

Be sure to answer the questions posed in the write-up and include your answers to the pre-lab.

- Last day of the experiment please fill out the [Experiment Evaluation](#)

## 13 The Custom VI's for this experiment

Henonbifr.VI Illustrates bifurcation in one of the variables of the Henon map.

Henon.VI Iterates the Henon map

NLSim.VI Displays time-series, return maps, and Fourier spectrum of many simulated signals and an arbitrarily acquired signal

PNJunc1.VI Acquires and analyzes time-series data acquired from the PN junction

NLDBifur.VI.

Quad.VI Iterates the quadratic map.

QuadBifr.VI Calculates the bifurcation diagram for the quadratic map.

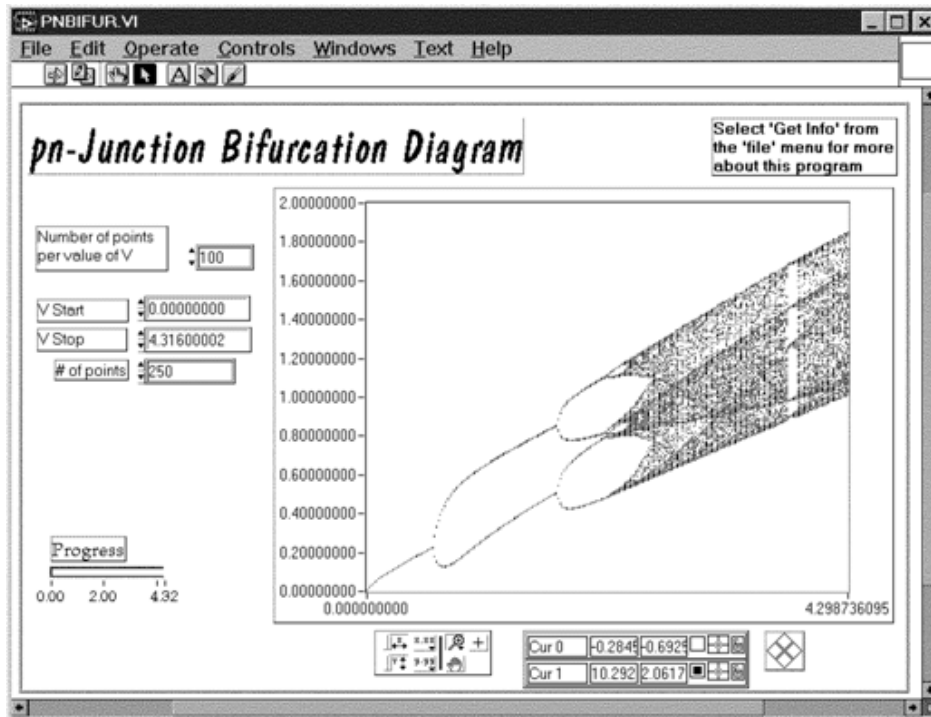


Figure 32: NLDBifur

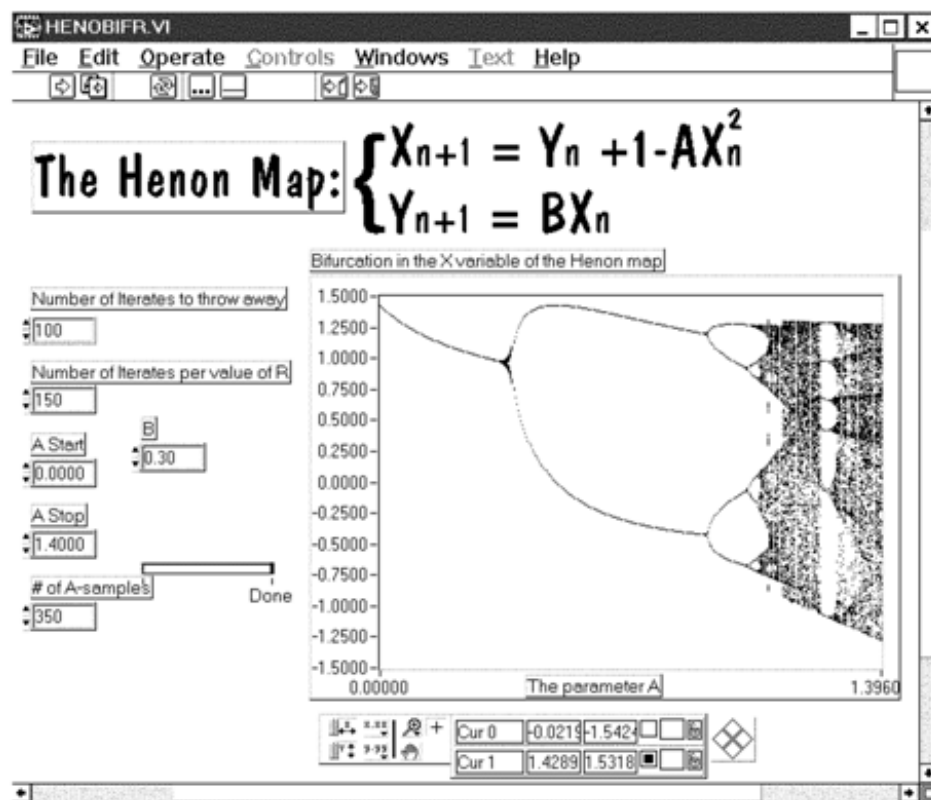


Figure 33: Henon Bifur

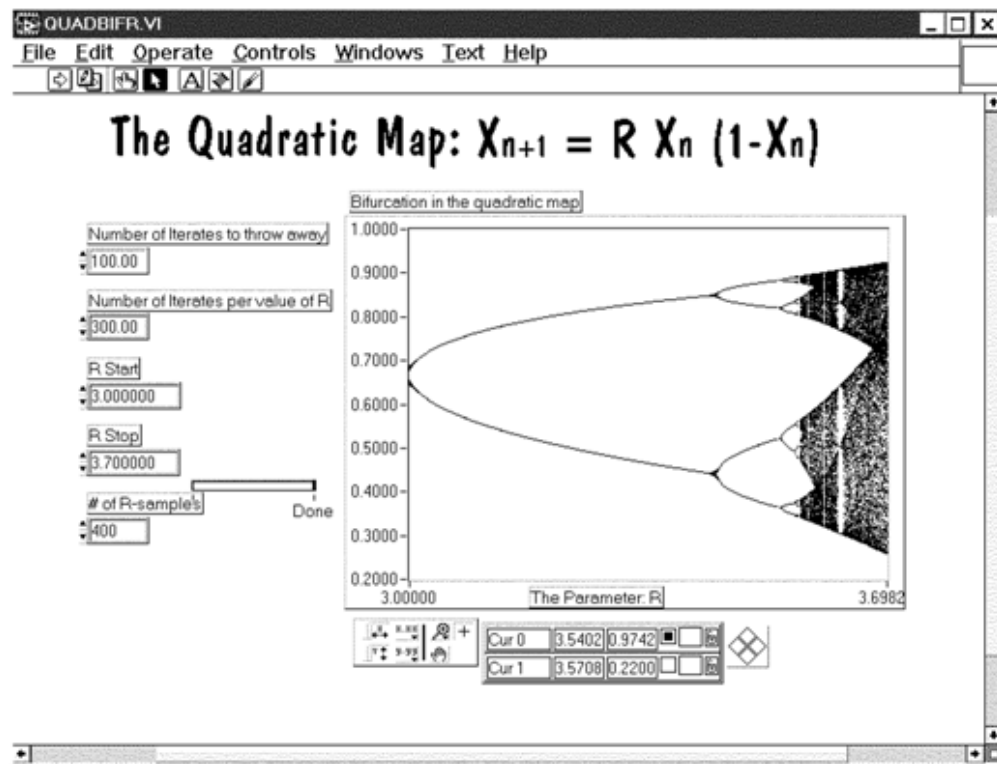


Figure 34: **Quad Bifur** This VI permits you to explore the Bifurcation diagram of the quadratic map. You can change the parameters to reveal the fine detail at the onset of chaos.

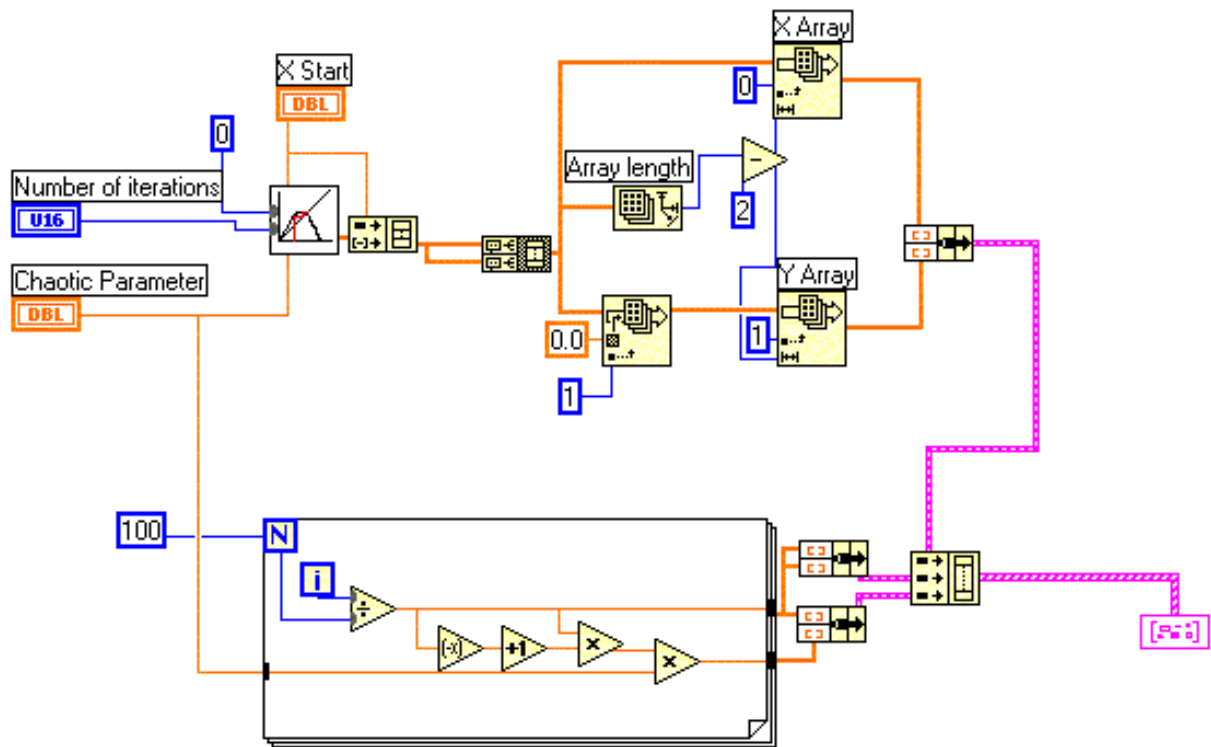


Figure 35: Cobweb analyzer for the quadratic map. **Cobweb2.vi**

Note: The elements in the current LabView program VI do not look quite like the elements in these diagrams.

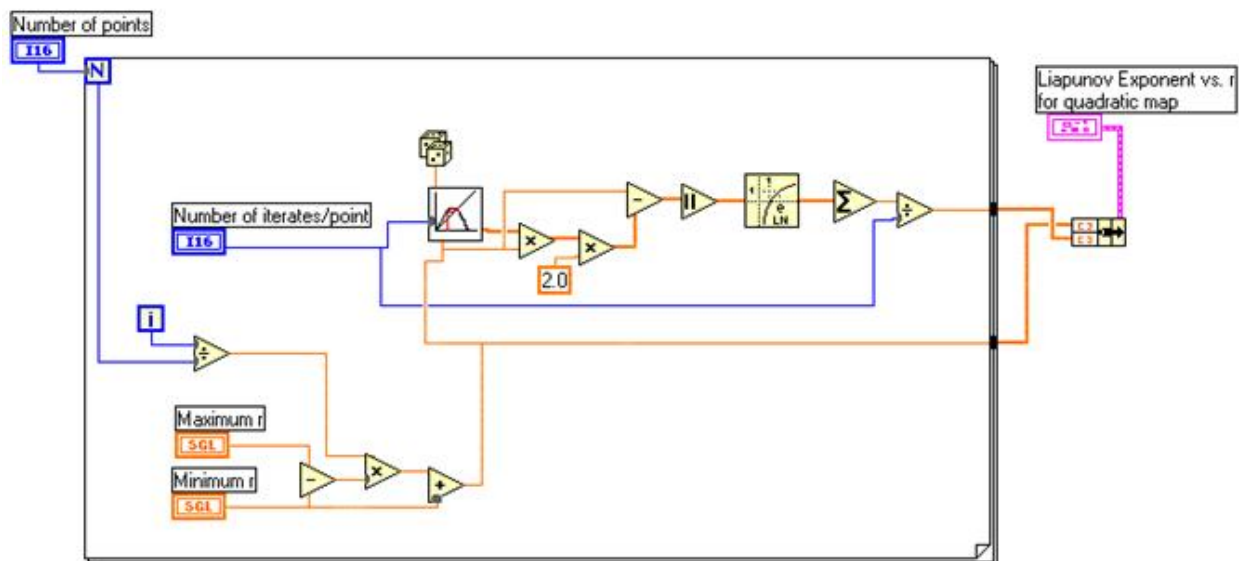


Figure 36: Liapunov exponent VI. **liapunov2.vi**