

Homework 6

CAAM 28200: Dynamical Systems with Applications

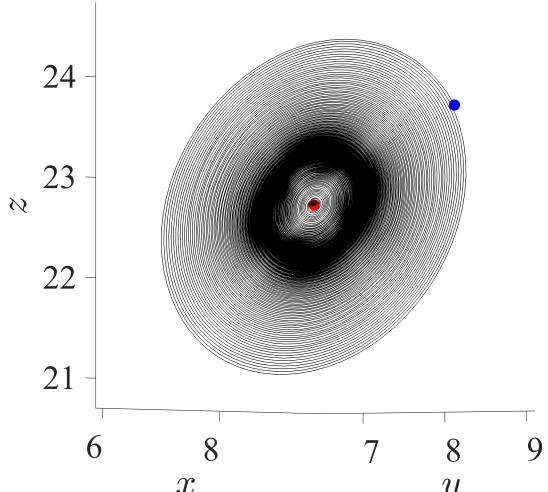
Kameel Khabaz

March 9, 2022

Simulation Exercises (Lorenz)

Here I simulate the Lorenz system for $\sigma = 10$, $b = 8/3$, and r varying from just below r_H to just above r_H . The resulting trajectories starting from x_0 near $C_+ = < \sqrt{b(r-1)}, \sqrt{b(r-1)}, r-1 >$ are shown below. We see here in the top row that when $r < r_H$, the trajectory spirals towards the stable fixed point C_+ . We can also tell this when we look at a plot of one of the coordinates versus time (on the left of the next figure). We see that while there is transient chaos at this value of r , the solution eventually damps down to equilibrium. On the other hand, when $r > r_H$, as on the bottom row, we clearly see chaotic solutions and we see the shape of the strange attractor. When we look at the plot of y vs. t , we clearly see that the solution for $r > r_H$ (on the right) is aperiodic.

$$r = 23.7368, r_H = 24.7368, \sigma = 10, b = 8/3$$



$$r = 25.7368, r_H = 24.7368, \sigma = 10, b = 8/3$$

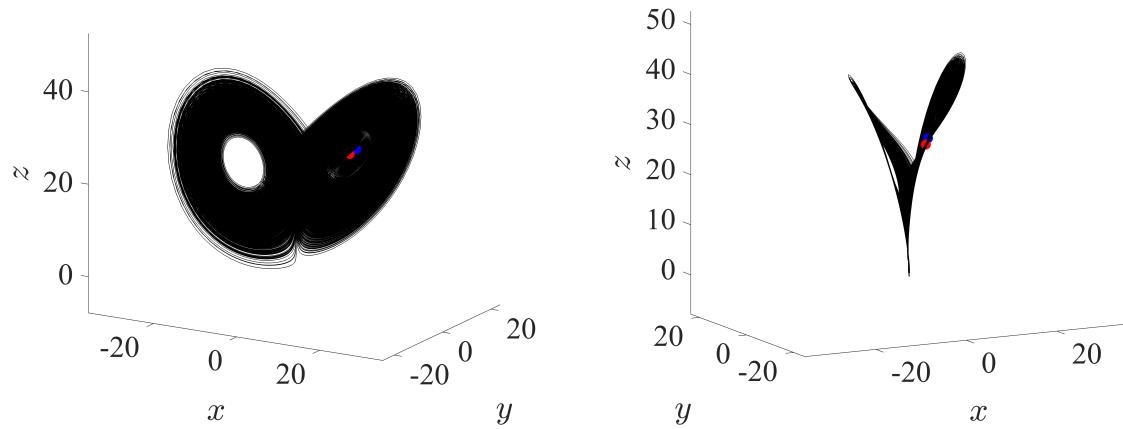


Figure 1: Lorenz Attractor

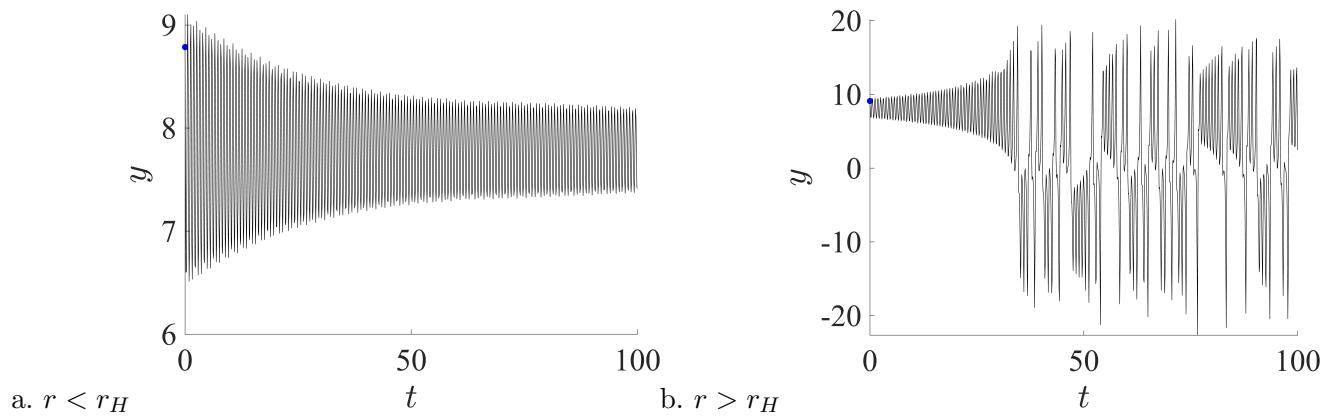


Figure 2: Lorenz Attractor

Now we start two trajectories with initial conditions x_0 and $y_0 = x_0 + \delta$ near x_0 . We see the result here, in which one starts at the blue dot and the other starts at the green dot. In this figure, we see that they must both be contained within the strange attractor over long time scales, but we cannot easily compare the trajectories in 3D.

$$r = 25.7368, r_H = 24.7368, \sigma = 10, b = 8/3$$

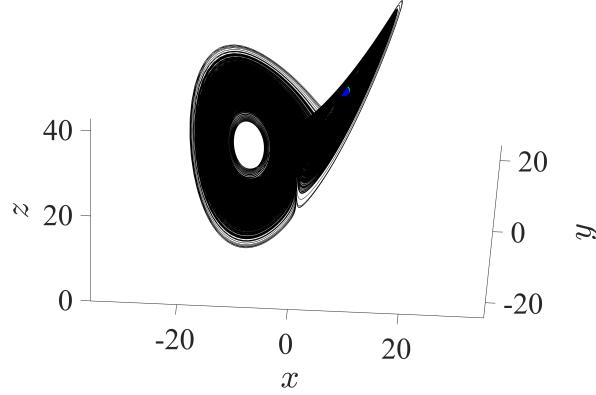


Figure 3: Lorenz Attractor: Two Trajectories

So, I plot the natural logarithm of the relative discrepancy between the trajectories $\ln\left(\frac{\|y(t)-x(t)\|}{\|\delta\|}\right)$ versus time. The plot is shown on the next page. We see here that regardless of δ , the relative discrepancy between the trajectory rapidly increases over time until it reaches a steady value when the two trajectories are mixed (operating relatively independently). The initial phase of exponential increase demonstrates the sensitive dependence on initial conditions of the Lorenz system, and the latter phase tells us that the solution for the Lorenz system is bounded to the order of the diameter of the strange attractor. We see that as δ decreases, the time range for exponential growth (which I zoom in on in the right panel) increases. So as the initial separation gets smaller, it takes longer and longer for the two trajectories to separate to the point where they operate independently.

When I found the growth rate in the log of the relative discrepancy, I was able to estimate the Lyapunov exponent. We know $\|\delta(t)\| \approx \|\delta_0\|e^{\lambda t}$, which implies $\ln \|\delta(t)\| \approx \lambda t + \ln \|\delta_0\|$. So, on the right panel, I calculate this linear fit and find that λ is around 0.03. We do see, however, that the value of λ varies slightly with the initial discrepancy. This $\lambda \approx 0.03$ is the largest Lyapunov exponent.

Based on this fit, we can calculate the time horizon before which predictions break down. We know that $t_{\text{horizon}} \approx O\left(\frac{1}{\lambda} \ln \frac{a}{\delta_0}\right)$. So, we predict that starting from $\delta = 10^{-6}$, the time that it takes for the discrepancy is greater than 10^{-1} is

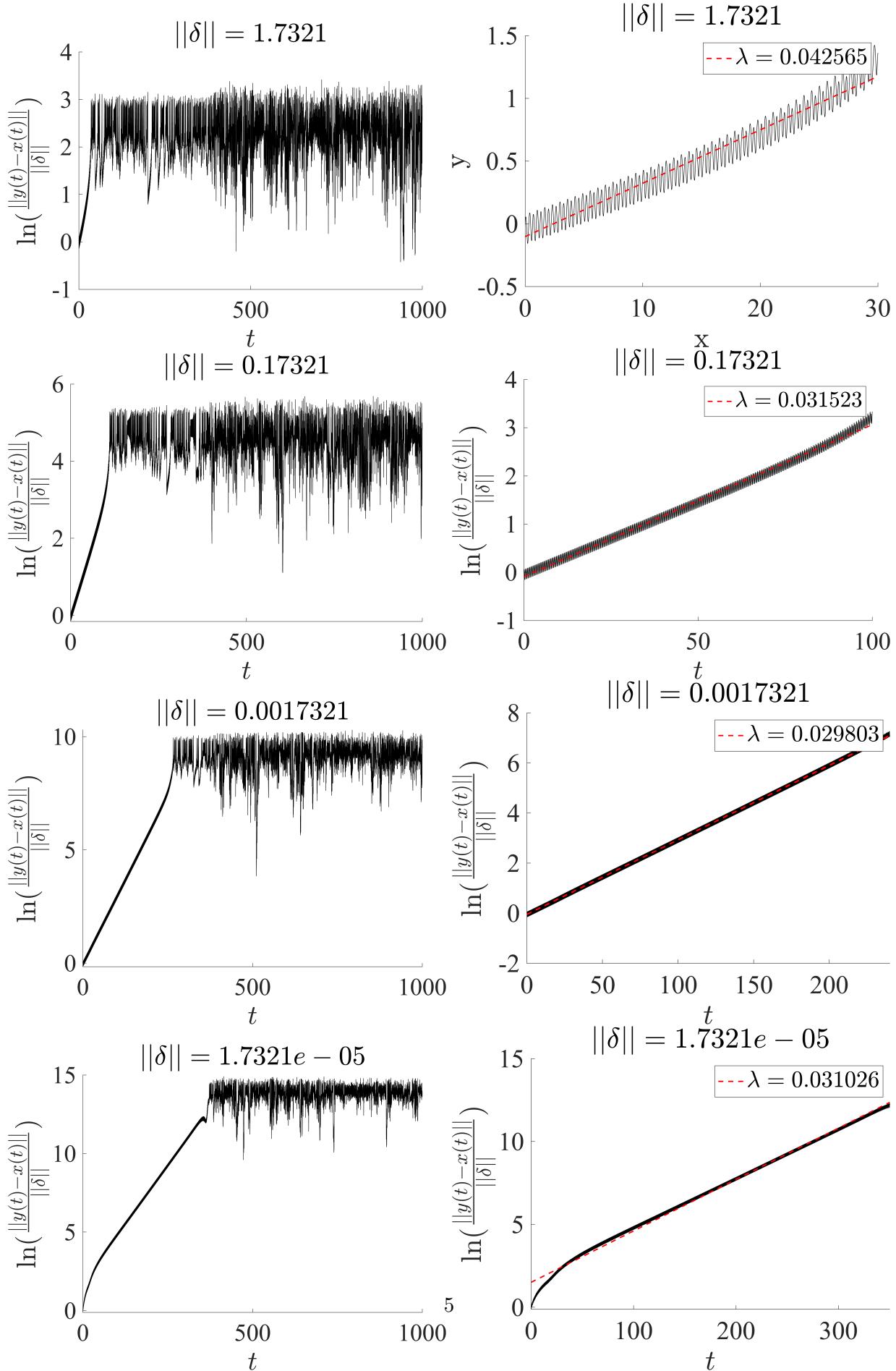
$$t_{\text{horizon}} \approx O\left(\frac{1}{\lambda} \ln \frac{a}{\delta_0}\right) = \frac{1}{0.03} \ln \frac{10^{-1}}{10^{-6}} = 384s$$

If we reduce our original measurement error dramatically such that $\delta = 10^{-9}$, then we find that

the new time horizon is

$$t_{horizon} \approx O\left(\frac{1}{\lambda} \ln \frac{a}{\delta_0}\right) = \frac{1}{0.03} \ln \frac{10^{-1}}{10^{-9}} = 614s$$

Thus, the new time horizon is only 1.6 times the old time horizon, or 60% longer. This minimal improvement in prediction capabilities given much higher measurement precision demonstrates the challenge of predicting chaotic systems.



Now, I record the sequence of local maxima of the $z(t)$, $\{z_1, z_2, \dots\}$, and make a scatter plot of z_{n+1} against z_n . This is the Lorenz Map, which we see below:

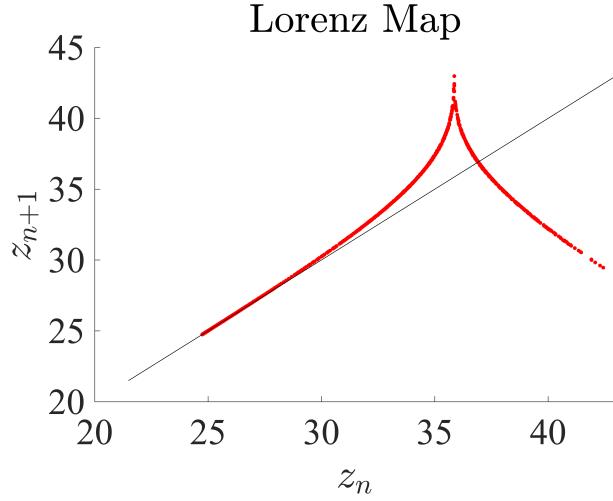


Figure 5: Lorenz Map

Here I plot some solutions to the Lorenz map using cobweb diagrams. I use a simple approximation of the curve in which I use the data value from the local maxima of $z(t)$ to tell me where z_{n+1} is (no interpolation). I obtain the following cobweb diagrams for different initial conditions. In the figure after that, I plot these cobweb diagrams with a lower density (smaller number of steps).

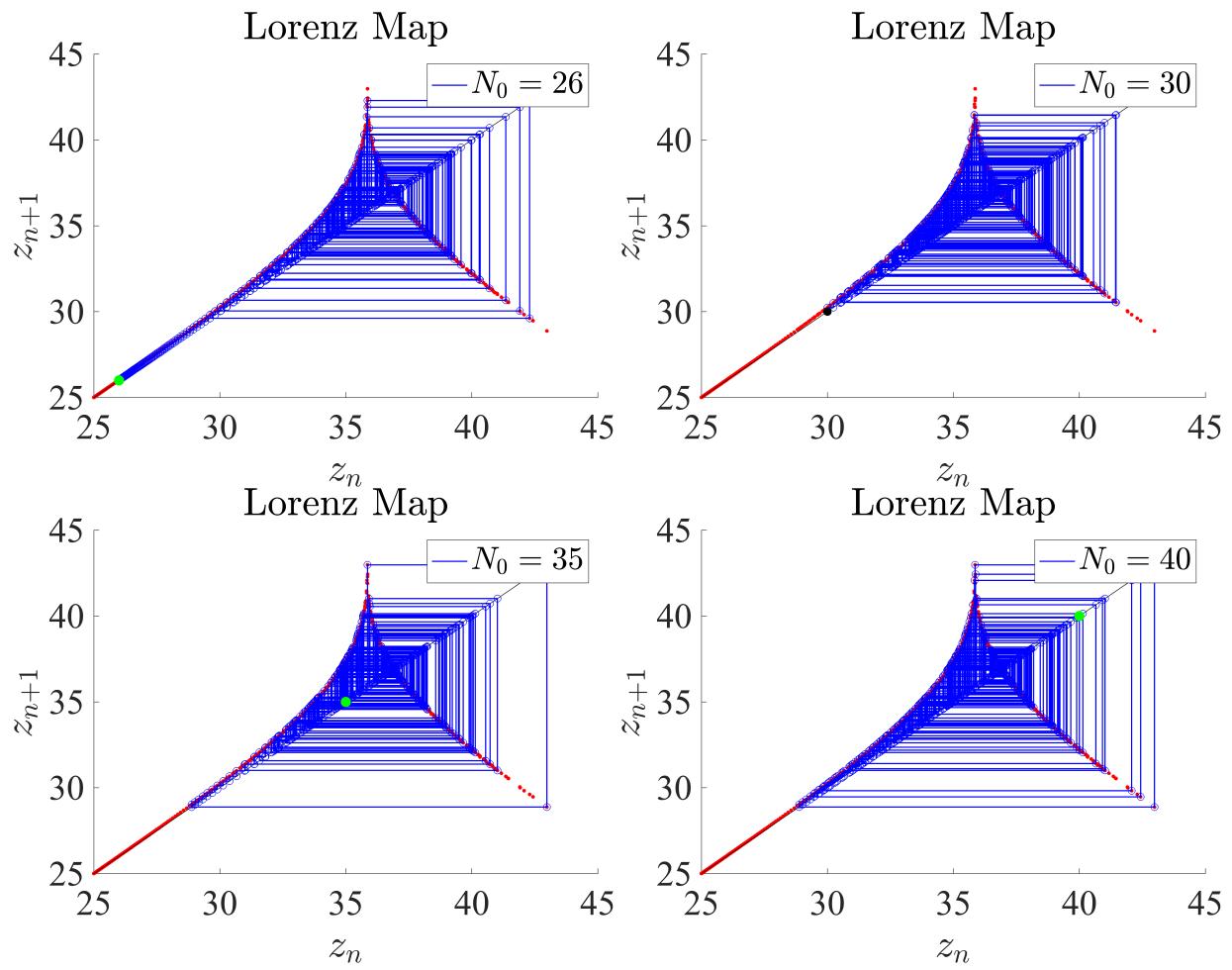


Figure 6: Lorenz Map Cobweb Diagrams, 500 Steps

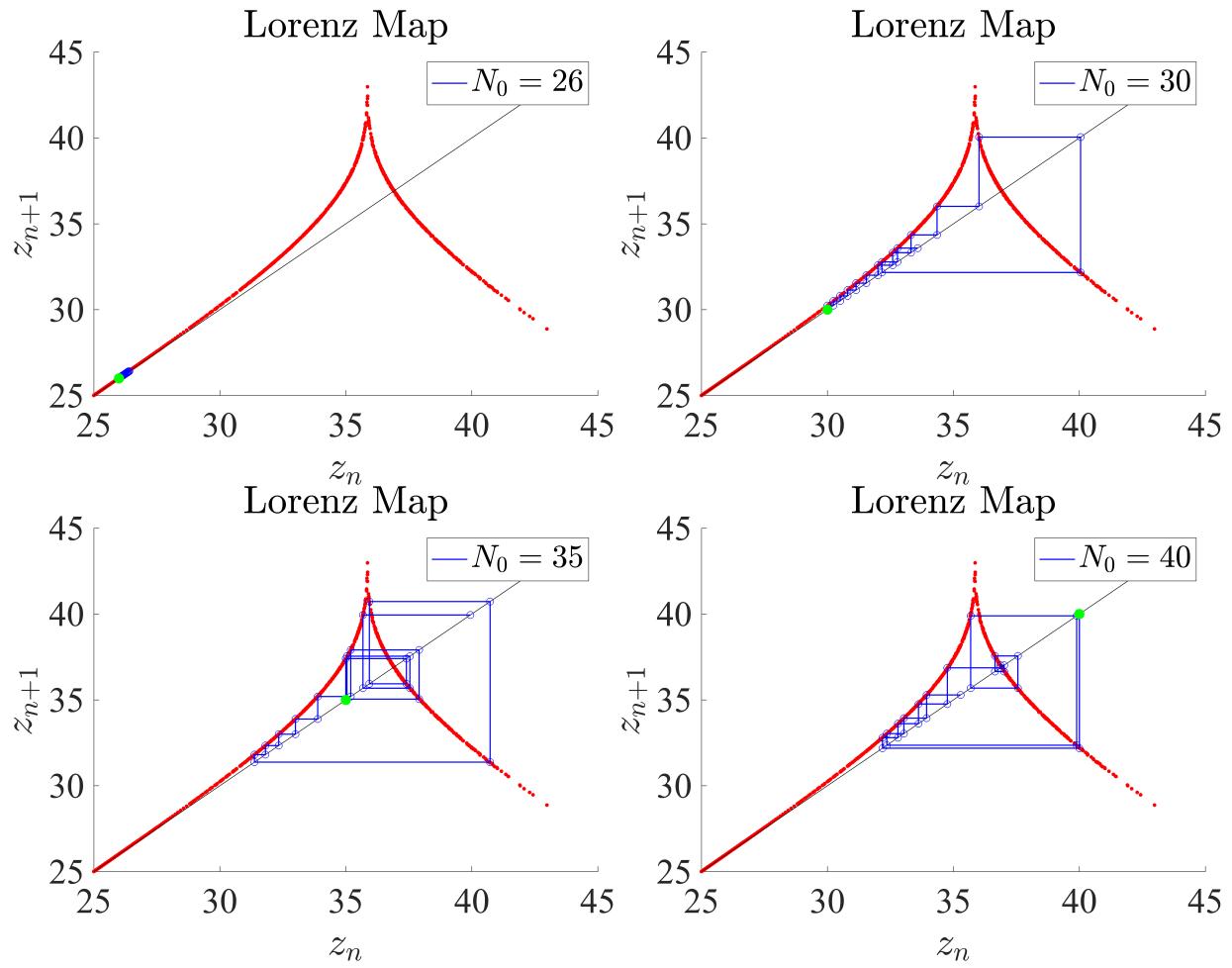


Figure 7: Lorenz Map Cobweb Diagrams, 30 Steps

Maps and Period Doubling

Here I run a recursion for the logistic map ($x_{n+1} = rx_n(1 - x_n)$) and the sine map ($x_{n+1} = r \sin(\pi x_n)$) for different values of r . For the logistic map, I sample r ranging from 3.4 to 4.0. For the sine map, I sample r ranging from 0.5 to 1.0. I then plot the last 1,000 points for each simulation versus the corresponding value of r to see where the map settles (as in an attractor). For both cases, we see an interesting structure in which the number of fixed points (or the number of lines) repeatedly doubles as r increases. When r reaches a critical value, it seems like there is no one fixed point and the solution becomes chaotic.

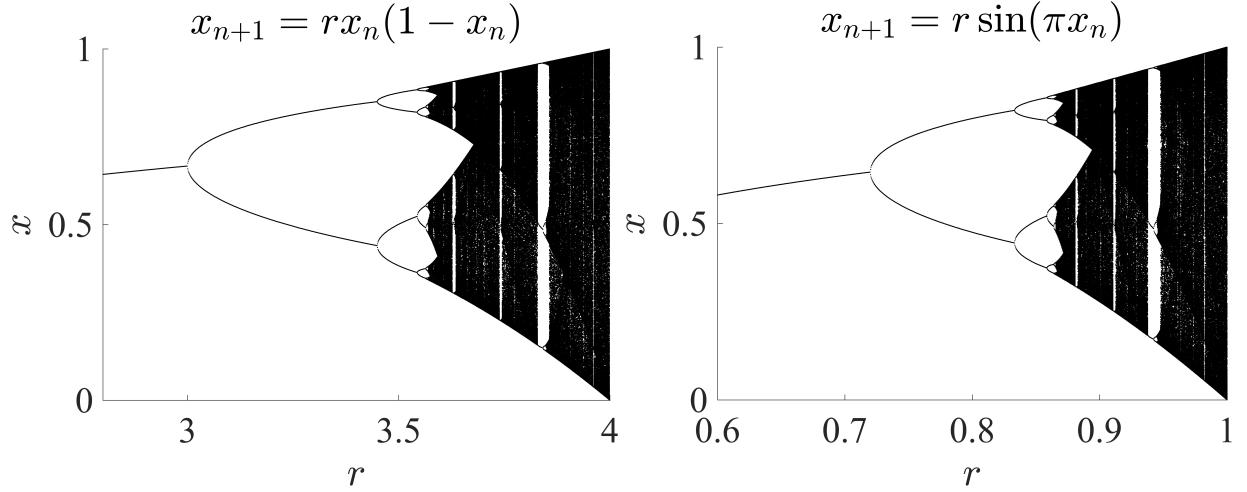


Figure 8: Logistic Map and Sine Map

If we zoom into the figure, we see that there is no singular attractor after r is above some critical value:

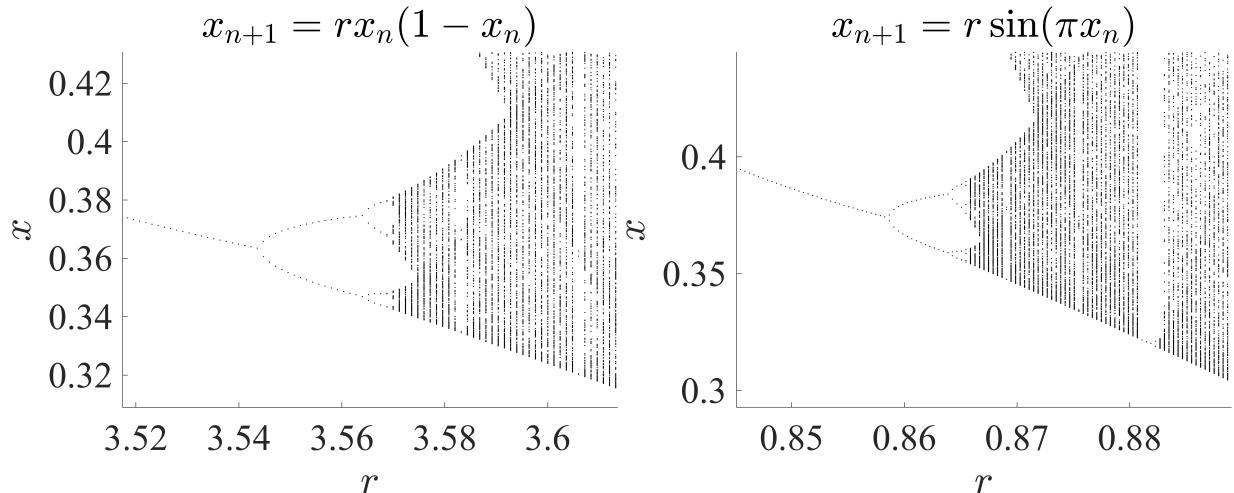


Figure 9: Logistic Map and Sine Map

The attractor also has a complex and interesting structure in which there seem to be small

vertical period windows that contain smaller version of the overall structure. If we zoom in to one of the period windows, we see the overall structure of the diagram reappear:

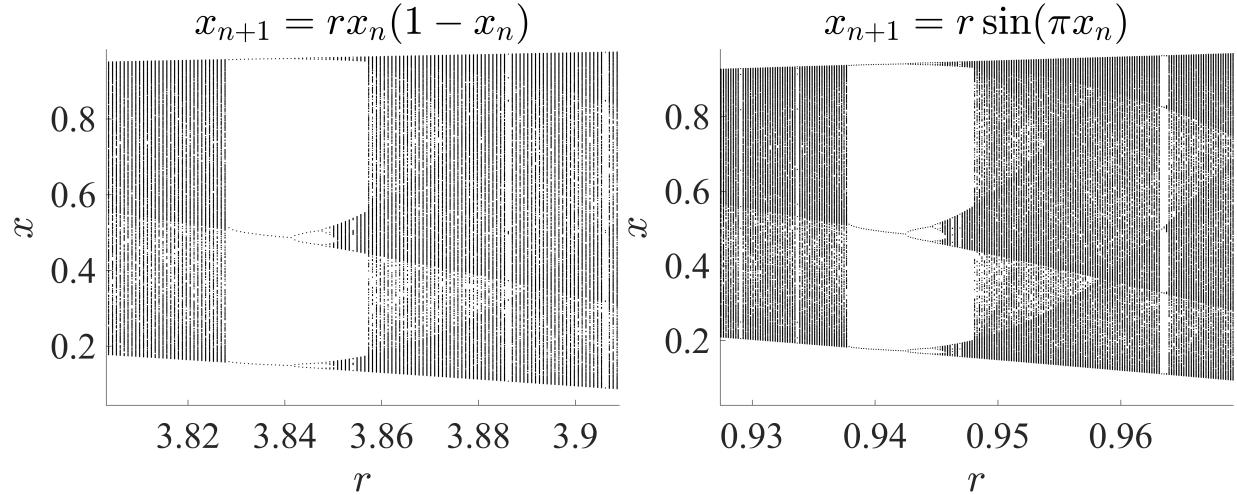


Figure 10: Logistic Map and Sine Map

I then adapted the code from the notes to calculate the values of r where the period doubles for the logistic system, obtaining $r_1 = 3.0000$, $r_2 = 3.4498$, $r_3 = 3.5443$, $r_4 = 3.5649$, $r_5 = 3.5690$. We know that from the Feigenbaum constant, we should expect that

$$\delta = \lim_{n \rightarrow \infty} \frac{r_n - r_{n-1}}{r_{n+1} - r_n} = 4.669\dots$$

This means that the spacing of the period doublings should approach a regular pattern in which $\Delta_n / \Delta_{n+1} \rightarrow \delta$. With my values, I obtained a δ relatively closed to that known value. For $n = 2$, I got $\delta = 4.7597$. For $n = 3$, I got $\delta = 4.5812$, and for $n = 4$, I got $\delta = 5.0263$. Increasing the number of periods and the numerical accuracy of my bifurcation detection would make my δ values better converge to the limit as n increases.

Code

```

1 % Kameel Khabaz
2 % CAAM 28200
3 % Homework 6 Lorenz Simulations
4
5 sigma = 10;
6 b = 8/3;
7 rH = (sigma * (sigma + b + 3))/(sigma - b -1);
8 r = rH - 1;
9 f = @myode;
```

```

10
11
12 figure()
13 close all;
14 hold on;
15 Cplus = [sqrt(b*(r-1)) sqrt(b*(r-1)) r-1];
16 initial_pos = [Cplus; Cplus + 1.* [1 1 1]];
17 for i = 1:2 %length(initial_pos)
18     scatter3(initial_pos(i,1),initial_pos(i,2),initial_pos(i,3),100,
19             'filled','b');
20     plot_solve(initial_pos(i,:),f,sigma, b, r)
21 end
22 scatter3(Cplus(1),Cplus(2),Cplus(3),100,'filled','r'); % C+ fixed
23 point
24 set(gca,'FontSize',30,'FontName','times')
25 xlabel("$x$",'Interpreter','latex')
26 ylabel("$y$",'Interpreter','latex')
27 zlabel("$z$",'Interpreter','latex')
28 set(gcf,'Position',[0 0 700 500])
29 title("$r = " + r + ", r_H = " + rH+ " , \sigma = " + sigma + ", b = "
30      8/3$",'Interpreter','latex')
31 %exportgraphics(gcf,"Lorenz_r" + r + ".png",'Resolution',600)
32 axis equal
33
34 %%
35 figure()
36 hold on
37 xlabel("$t$",'Interpreter','latex')
38 ylabel("$y$",'Interpreter','latex')
39 for i = 2 %length(initial_pos)
40     scatter(0,initial_pos(i,2),'filled','b');
41     tspan = 0:.1:100;
42     [~,sol] = ode45(@(t,y) myode(t,y,sigma, b, r),tspan,initial_pos(
43         i,:));
44     plot(tspan, sol(:,2),'k','LineWidth',.5);
45 end
46 set(gca,'FontSize',30,'FontName','times')
47
48 %% 2 neighboring trajectories

```

```

45 sigma = 10;
46 b = 8/3;
47 rH = (sigma * (sigma + b + 3))/(sigma - b -1);
48 r = rH + 1;
49 f = @myode;
50
51
52 figure()
53 close all;
54 hold on;
55 delta = [.001 .001 .001];
56 Cplus = [sqrt(b*(r-1)) sqrt(b*(r-1)) r-1];
57 initial_pos = [Cplus; Cplus + delta];
58 cols = ['b' 'g']
59 solutions = {};
60 for i = 1:2 %ength(initial_pos)
61     y0 = initial_pos(i,:);
62     scatter3(initial_pos(i,1),initial_pos(i,2),initial_pos(i,3),100,
63               'filled',cols(i));
64     tspan = 0:.01:1000;
65     opts = odeset('RelTol',1e-7,'AbsTol',1e-6);
66     [~,sol] = ode45(@t,y) myode(t,y,sigma, b, r),tspan,y0,opts);
67     solutions{end+1} = sol;
68     plot3(sol(:,1), sol(:,2),sol(:,3), 'k', 'LineWidth', .5);
69 end
70 set(gca,'FontSize',30,'FontName','times')
71 xlabel("$x$",'Interpreter','latex')
72 ylabel("$y$",'Interpreter','latex')
73 zlabel("$z$",'Interpreter','latex')
74 set(gcf,'Position',[0 0 700 500])
75 title("$r = " + r + ", r_H = " + rH+ " , \sigma = " + sigma + ", b = " + b + " , " + "8/3$",'Interpreter','latex')
76 %exportgraphics(gcf,"Lorenz_r" + r + "2_trajectories.png",'Resolution',600)
77 axis equal
78
79 % Discrepancies plot
80 figure()

```

```

81 hold on;
82 discrep = vecnorm(solutions{1} - solutions{2},2,2);
83 normdiscrep = discrep ./ norm(delta);
84 normlogd = log(normdiscrep);
85 plot(tspan',normlogd,'k')
86 %{
87 % Now do the fit
88 linidcs = 1:find(tspan == 350);
89 f = fit(tspan(linidcs)',normlogd(linidcs),'poly1');
90 h = plot(f);
91 h.LineStyle = '--';
92 h.Color = "r";
93 h.LineWidth = 1.5;
94 legend(h,"$\lambda = "+f.p1+"$",'Interpreter','latex')
95 %}
96 xlabel("$t$",'Interpreter','latex')
97 ylabel("$\ln(\frac{|y(t) - x(t)|}{|\delta|})$",'Interpreter','
    latex')
98 title("$|\delta| = " + norm(delta) + "$",'Interpreter','latex')
99 set(gca,'FontSize',30,'FontName','times')

100
101 %% Find local maxima
102 z = sol(:,3);
103 lmax = islocalmax(z,'MinProminence',.01);
104 map = [[0; z(lmax)] z(lmax); 0] ];
105 map = map(2:end-1,:);
106 x = [max(map(:))/2:.1:max(map(:))];
107 y = x;
108 figure()
109 hold on
110 scatter(map(:,1),map(:,2),10,'r','filled')
111 plot(x, y,'k')
112 set(gca,'FontSize',30,'FontName','times')
113 xlabel("$z_n$",'Interpreter','latex')
114 ylabel("$z_{n+1}$",'Interpreter','latex')
115 title("Lorenz Map",'Interpreter','latex')
116 exportgraphics(gcf,"Lorenz_Map.png",'Resolution',600)
117 %% Make cobweb plot
118 ts = linspace(0,1,30);

```

```

119
120
121 f1 = figure()
122 hold on
123 scatter(map(:,1),map(:,2),10,'r','filled')
124 plot(x, y,'k')
125 set(gca,'FontSize',30,'FontName','times')
126
127
128 xlabel("$z_n$",'Interpreter','latex')
129 ylabel("$z_{n+1}$",'Interpreter','latex')
130 xlim([25 45])
131 ylim([25 45])
132 set(gca,'FontName','Times','FontSize',30)
133
134 Ns = zeros(length(ts),1);
135 No = 35;
136 Ns(1) = No;
137 for i = 2:length(ts)
138     if mod(i,2) == 1
139         [~,mini] = min(abs(map(:,1) - Ns(i-1)));
140         Ns(i) = map(mini,2);
141     else
142         Ns(i) = Ns(i-1);
143     end
144 end
145 Ns_p1 = Ns(2:end);
146 Ns = Ns(1:end-1);
147 h = plot(Ns,Ns_p1,'b','LineWidth',1)
148 scatter(Ns,Ns_p1,'b')
149 scatter(No,No,75,'g','filled')
150 title("Lorenz Map",'Interpreter','latex')
151 legend(h,"$N_0 = " + No + "$",'Interpreter','latex')
152 exportgraphics(gcf,"Lorenz_map_cobwebs_No" + No + "_ld.png",
    'Resolution',600)
153
154
155 %% Maps and Period Doubling
156
```

```

157 logistic = @(r,x) r .* x .* (1 - x);
158 sine = @(r, x) r .* sin(pi .* x);
159 xo = 0.5;
160 rs = linspace(3.6,4,500);
161
162 % Do logistic
163 figure()
164 hold on
165 for i = 1:length(rs)
166     logsols = recursion(logistic, rs(i), 100000, xo);
167     scatter(repmat(rs(i), 1000,1), logsols(end-1000+1:end),1,'k','filled')
168 end
169 set(gca,'FontName','Times','FontSize',30)
170 xlabel("$r$",'Interpreter','latex')
171 ylabel("$x$",'Interpreter','latex')
172 title("$x_{n+1} = rx_n(1-x_n)$",'Interpreter','latex')
173 %exportgraphics(gcf,"Logistic_map.png","Resolution",600)
174
175 %% Do sine map
176 figure()
177 hold on
178 rs = linspace(.9,1,500);
179 for i = 1:length(rs)
180     logsols = recursion(sine, rs(i), 100000, xo);
181     scatter(repmat(rs(i), 1000,1), logsols(end-1000+1:end),1,'k','filled');
182 end
183 set(gca,'FontName','Times','FontSize',30)
184 xlabel("$r$",'Interpreter','latex')
185 ylabel("$x$",'Interpreter','latex')
186 title("$x_{n+1} = r\sin(\pi x_n)$",'Interpreter','latex')
187 %exportgraphics(gcf,"Sine_map.png","Resolution",600)
188
189 %% Calculating r of bifurcations, adapted from Strang notes
190 rs = linspace(2.8,3.57,10^4); % Increase spacing of r
191 xs = linspace(0,1,10^4);
192 f_logistic = @(x,r) r*x.* (1 - x);
193 f_sine = @(x,r) r*sin(pi*x);

```

```

194
195 f{1} = @(x,r) f_logistic(x,r);
196 for j = 1:31
197     f{j+1} = @(x,r) f{1}(f{j}(x,r),r);
198 end
199
200 new_period = 0;
201 rBifurcations = [];
202 for j = 1:length(rs)
203     r = rs(j);
204     % find fixed points and cycles (up to period 16)
205     period = new_period;
206     for p = 0:period
207         step = 2^p;
208         % find intersections
209         sign_func = @(x) sign(f{step}(x,r) - x);
210         signs = sign_func(xs);
211         indices{p+1} = find(abs(diff(signs)) ~= 0); % indices
212             % corresponding to equilibrium for period 2^p
213         indices{p+1} = sort(indices{p+1}, 'ascend');
214         indices{p+1}(1) = []; % drop the equilibrium at 0
215         % get stability
216         stability = nan(size(indices{p+1}));
217         for index_n = 1:length(indices{p+1})
218             index = indices{p+1}(index_n);
219             slope = (f{step}(xs(index+1),r) - f{step}(xs(index-1),r)
220                     )/(xs(index+1) - xs(index - 1));
221             stability(index_n) = abs(slope);
222         end
223         unstable{p+1} = indices{p+1}(stability > 1);
224         stable{p+1} = indices{p+1}(stability < 1);
225     end
226     % check if we need higher period
227     if length(unstable{period+1}) == length(indices{period+1})
228         new_period = period+1;
229         rBifurcations = [rBifurcations r];
230     end
231     % bound to avoid diverging

```

```

231     new_period = min(new_period,7);
232
233 end
234
235 function sols = recursion(f, r, nsteps, xo)
236     sols = nan(nsteps,1);
237     sols(1) = xo;
238     for i = 2:nsteps
239         sols(i) = f(r, sols(i-1));
240     end
241 end
242
243 function dydt = myode(~,yvec,sigma,b,r)
244     x = yvec(1);
245     y = yvec(2);
246     z = yvec(3);
247     xdot = sigma * (y - x);
248     ydot = r * x - y - x .* z;
249     zdot = x .* y - b * z;
250     dydt = [xdot; ydot; zdot];
251 end

```