

Directions: Use only **ONE** side of each page, use ink and a staple. This lab is about chaos. The kind of chaos we are looking at is called sensitivity to initial conditions.

The starting point is the Lorenz equations

$$\begin{aligned}\frac{dx}{dt} &= -\sigma x + \sigma y \\ \frac{dy}{dt} &= rx - y - xz \\ \frac{dz}{dt} &= -bz + xy\end{aligned}$$

These are given on page 559 of the textbook. We will use the values $\sigma = 10, r = 28, b = 8/3$. A vague description of the model is in the text. The behavior of the system is more important than the model as the system has become the poster child for chaos.

We are going to use scilab to solve the Lorenz system using several different (but very close) initial values. Our **reference solution** has the initial values of $x(0) = 5, y(0) = 10, z(0) = 10$, the other solutions will only change the initial value of $x(0)$ by a positive Δx .

1. Use scilab to solve the Lorenz system for the initial value problem $x(0) = 5, y(0) = 10, z(0) = 10$ (our reference solution with $\Delta x = 0$) and for the every so slightly different initial values $x(0) = 5.01, y(0) = 10, z(0) = 10$ (with $\Delta x = 0.01$). Plot x vs t and obtain a plot very much like Figure 9.9.3 on page 562. For this plot a t range of $t = 0 : 0.1 : 20$ works nicely.

Deliverables

- (a) The plot above, with both curves on the same plot.
 - (b) Looking at the two graphs there 3 regions. In beginning the two data points are so close that the graph can't tell them apart. (Let's agree to say points within 0.150 are too close to tell apart. The selection of this value is given in the appendix.) Then at some point $T_1 > 5$ the plot separates the data points from the two curves but the curves are continuing to be almost same until some point $T_2 < 10$ where the two curves have a radial parting of of the ways. Find T_1 and T_2 .
2. Your second job is to plot the solution in xyz -space. The new command you will need in scilab is

```
-->param3d(x,y,z);
```

and it will plot the parameteric equations $x = x(t), y = y(t), z = z(t)$. The default 3D window has $\alpha = 35$ and $\theta = 45$ which does not do justice to the figure. There is a 3D Rot. button which allows the user to rotate the figure in 3D on the fly. Only the bounding box rotates in real time, a second click will redraw the figure. Once you have found a pleasing picture of the butterfly, note the alpha and theta values at the top of the window. You can force the rotation by using the slightly different param3d command, below 45 is the θ -value, 35 is the α -value. The angles α and θ are related to the observation point in the spherical coordinates ϕ and θ that you learned in Calculus 3.

```
-->param3d(x,y,z,45,35);
```

The help page for param3d plot shows more options if you are interested.

Deliverables

- (a) Two 3D plots of reference solution. One with $\Delta t = 0.1$ and another with $\Delta t = 0.01$, both with the same α and θ selected to to be a pleasing picture of our butterfly.
- (b) Why does the $\Delta t = 0.1$ look the way it does and why does the $\Delta t = 0.01$ fix the problem?

- (c) Our butterfly is basically two flat wings make up of spirals. Do the trajectories spiral in or spiral out? (Part 3 might help you with this question.)
3. Use a phase plane plot in the yz -plane to zero in the divergence of the reference solution with $\Delta x = 0$ and the nearby solution with $\Delta x = 0.01$. Plot the time period from $t = 5.0$ to $t = 9.5$ and put a marker on quarter second. [If w is the reference solution for $\Delta t = 0.01$ and hence $t = 0 : 0.01 : 20$ then $w(2, 501 : 25 : 951)$ will give the quarter second points. (See the sample phase code below.)

Deliverables

- (a) The plot mentioned above, with both curves on the same plot.
- (b) Use the plot to help you answer, for nearby plots is it the inner or is it the outer trajectory that moves fastest?
4. Our final item is to vary Δx by powers of 0.1 and consider the x value of five perturbed solutions with $\Delta x = 0.1, 0.01, 0.0001, 0.00001, 0.000001$. Again a range of $t = 0 : 0.1 : 20$ is used so that we can reuse the work from Part 1. We need to eyeball the values for T_1 and T_2 (defined in Part 1) for each of these solutions and fill in a table.

Deliverables

- (a) One plot of the five differences $x_a - x_r$ where $x_a(t)$ is the x value of the approximate solution for one of the given Δx and $x_r(t)$ the x value of the reference solution.
- (b) Complete the table

Δx	T_1	T_2
0.1	?	?
0.01	from Part 1	from Part 1
0.001	?	?
0.0001	?	?
0.00001	?	?

Sample phase code

```
--> t=0:%pi/1200:4*%pi; w=[t.*cos(t); t.*sin(t)];
--> clf(); plot(w(1,:),w(2,:),"b"); // spiral for 0 <= t <= 4 pi, 2 turns
--> s=2401:200:4801; // 2 pi <= t <= 4 pi by pi/6's
--> plot(w(1,s),w(2,s),"bo.."); // just the points no connecting lines
```

Appendix: How close is the same on a graph (informational only, not required)

Here is a sample back of the envelop calculation of how far apart two data points can be and still be the same point on a computer screen. Scilab drew a 610x461 window (in pixels) for a graph for part I. The vertical scale was about 2/3 of the window and represented a range of -20 to 20. So roughly 300 pixels for a range of 40, each pixel covers 40/300 or roughly 0.133 of a unit. So data points less than 0.1 often graph to the same pixel for this particular window and plot, while data points more than 0.14 would be in different pixels. Maximizing the window on my big screen changes the values to window has height 1107, so roughly 750 pixels for a range of 40 so each pixel covers 40/750 or roughly 0.053 of a unit.