Math 215 El Niño Southern Oscillations Homework

Nathan Bailey, Sven Bachmann, and Peter Harrington

1 A dynamical system for El Niño

El Niño is a quasi-periodic phenomena in which the sea surface temperature (SST) becomes much colder than usual in the tropical Eastern Pacific. In fact, there are oscillations between El Niño and La Niña, the latter corresponding to a warmer than average SST in the tropical Eastern Pacific. One of the first models to describe the El Niño system considers the effect of trade winds on the interaction between the SST and the depth of the transition layer (called the thermocline) between the warm surface and the cold ocean floor. The model is summarized in the following system of ODEs:

$$\begin{cases} x' &= -x + \gamma(bx+y) - \varepsilon(bx+y)^3 \\ y' &= -ry - \alpha bx \end{cases} \tag{1}$$

where x is proportional to the *SST anomaly* in the Eastern Pacific (namely the difference between the warm SST during El Niño and standard SST), and y is proportional to the *thermocline depth anomaly* in the Western Pacific. All geophysical parameters $\gamma, b, \epsilon, r, \alpha$ are strictly positive.

The purpose of this module is to study the oscillatory solutions of the system (1), both in the linear and non-linear regimes.

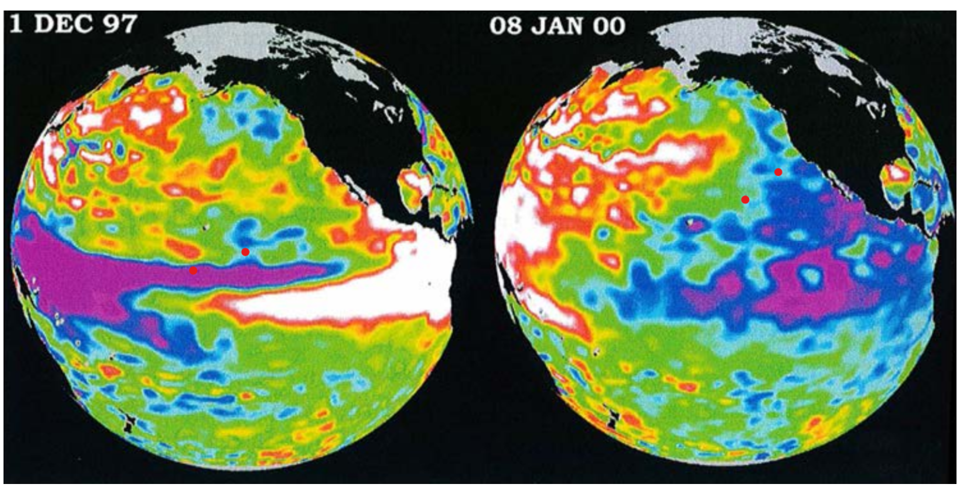

Figure 1: Temperature measurements illustrating the oscillation between El Niño (left) and La Niña (right) (Kapler and Engler, reprinted from NASA/JPL-Caltech)

Note for instructors: Below we present two options for questions: one which contains all the parameters in the model and one which contains only one parameter. We present the questions using the full parameter set first. Note that for the questions with only one parameter, the rest of the parameter values are chosen so they line up with the computational part of the assignment. If the computational part of the assignment is not being used, simpler parameter values could be chosen.

2 Questions (including all parameters)

The El Nino system can be represented by the following ODEs:

$$x' = -x + \gamma(bx + y) - \varepsilon(bx + y)^3 \tag{2}$$

$$y' = -ry - \alpha bx \tag{3}$$

where x is proportional to the Sea Surface Temperature (SST) anomaly in the Eastern Pacific (the difference between the warm SST during El Nino and standard SST), and y is proportional to the thermocline depth anomaly in the Western Pacific.

1. Linearize the system about the fixed point $(0, 0)$ using the Jacobian. Find restrictions on the parameters so that solutions to the linearized system are purely oscillatory. Find the eigenvalues of the system in this case (they should be complex conjugates of each other and have zero real part). Hint: it might be useful to group $\gamma b = c$ once you linearize).

2. When we observe pure oscillations, the solution set will have two complex conjugate eigenvalues $\lambda_1, \lambda_2 = \pm i\sqrt{\text{Det}(A)}$. Find a general solution for this equation using the eigenvectors for these given eigenvalues.

3 Questions (with all but one fixed parameter values)

These two phenomena can be represented in a system of ODE's:

$$x' = -x + \frac{3}{4}(bx + y) - 0.1(bx + y)^3 \tag{4}$$

$$y' = -\frac{1}{4}y - \frac{b}{8}x \tag{5}$$

where x is proportional to the Sea Surface Temperature (SST) anomaly in the Eastern Pacific (the difference between the warm SST during El Nino and standard SST), and y is proportional to the thermocline depth anomaly in the Western Pacific.

1. Linearize the system about the fixed point $(0, 0)$ using the Jacobian. Find a restriction on the b so that solutions to the linearized system are purely oscillatory. Find the eigenvalues of the system in this case (they should be complex conjugates of each other and have zero real part). Hint: once you find one restriction required for purely complex eigenvalues, use it to simplify the calculation of the eigenvalues.

2. When we observe pure oscillations, the solution set will have two complex conjugate eigenvalues $\lambda_1, \lambda_2 = \pm i\sqrt{\text{Det}(A)}$. Find a general solution for this equation using the eigenvectors for these given eigenvalues. Use the value of b which you found in the previous problem.

4 Numerics assignment (for syzygy)

(.ipynb file can be found in folder. If you make changes to ipynb file, you can run ‘jupyter nbconvert –to latex 215ENSOnumerics.ipynb’ in the syzygy terminal to export assignment to latex)

ENSO assignment

The El Nino Southern Oscilllations are an observed phenomenon in which the average tropical Eastern Pacific Sea Surface Temperature becomes much colder than usual. This oscillation can be represented in a system of two nonlinear ODE’s, with the following two variables: x which is proportional to the Sea Surface Temperature (SST) anomaly in the tropical Eastern Pacific, and y which is proportional to the thermocline depth anomaly in the Western Pacific. The thermocline is the (rough) boundary between the warmer top layer of the ocean and the cooler deep ocean layer.

Consider the following nonlinear system of ODEs:

$$x' = -x + \gamma(bx + y) - \varepsilon(bx + y)^3 \tag{6}$$
$$y' = -ry - \alpha bx \tag{7}$$

For this system, first find the linearization about the origin (by hand), then follow the instructions on how to use plt.streamplot and plot the linearization about the origin.

These statements import the necessary tools to plot our equation:

[5]:
```
from matplotlib.pyplot import cm
import matplotlib.pyplot as plt
import numpy as np
import math
```

The constants below determine the behavior of the system around the origin. Feel free to change these parameters and see what happens to the linearization.

[6]:
```
g = 3/4 #gamma
b = 5/3
e = 0.1 #epsilon
r = 1/4
a = 1/8 #alpha
```

The following code segment sets up a meshgrid (i.e. small rectangular chunks that we can solve the equation in) The bounds in np.arrange describe the size of the meshgrid, or the domain over which we’re solving.

[7]:
```
nx, ny = .1, .1
x = np.arange(-2, 2, nx)
y = np.arange(-2, 2, ny)
X, Y = np.meshgrid(x, y)
```

Next, add your linearized system below: e.g.

dx = -x + ...

dy = -r*y -...

[4]:
```
dx =
dy =
```

The block below plots the system for the given differential equations using different points on the meshgrid as initial conditions to computationally solve the system. This is essentially a slope field, but with more connected trajectories. Use the information provided here to run the command plt.streamplot.

[ ]:
```
plt.streamplot(X,Y,dx, dy, density=2, cmap= 'jet', arrowsize=1)
```

Now add your solution to the linearized system from the written assignment to the code block below. When you run the code block, your solution will be overlaid overtop the slope field. Modify c_1 and c_2 so that the range of your solution in x is approximately between -0.5 and 0.5. Because you are not required to start at a specific initial condition, you may set one of your constants to zero and only modify the other constant. Keep the parameters constant at this point onwards.

[ ]:
```
#Note: use np.cos(t) for cos(t), np.sin(t) for sin(t), and np.sqrt() to sqrt()
# use x**2 to enter x^2
g = 3/4 #gamma
b = 5/3
e = 0.1 #epsilon
r = 1/4
a = 1/8 #alpha

c_1 =
c_2 =

w_c = #you can define omega here in terms of the other parameters if you'd like, it␣
↪may make it easier to enter your solution

t=np.linspace(0,50,100)

def x(t):
    return #add your function before this comment

def y(t):
    return #add your function before this comment

plt.streamplot(X,Y,dx, dy, density=2, cmap= 'jet', arrowsize=1)
plt.plot(x(t),y(t))
```

The code below plots the slope field for the full non-linear system and overlays your linearized solution on top. It also plots one trajectory of the numerical solution to the non-linear equation so that you can see how well the non-linear solution aligns with your linearized solution.

What you see is the existence of a stable limit cycle in the non-linear system, this means that all trajectories approach the cycle that is approximated by your linearized solution.

Note: You can go back to the code box above and edit your c_1 and c_2 if you want to change the amplitude of your linearized solution so that it lines up well with the numerical solution to the non-linear system.

[10]:
```
#First, find numerical solution to full non-linear system
#Set b=5.3/3. This is because at b=5/3 the fixed point is actually stable, so we need␣
↪to increase b a little to generate a limit cycle that will match with the linearized␣
↪solution
b=5.3/3
from scipy.integrate import solve_ivp
def thc(t,z,a,b,e,g,r):
    x,y=z
```

```python
    return [-x + g*(b*x + y) - e*(b*x + y)**3, -r*y - a*b*x]

t_eval = np.linspace(0,100,500)

# Here the [1,1] part of the code specifies the initial conditions. Feel free to
 ↪change them
sol = solve_ivp(thc, [0, 100], [1, 1], t_eval=t_eval, args=(a,b,e,g,r), max_step=0.01)

#plot slopefield:
dx = -X + g*(b*X + Y) - e*(b*X + Y)**3
dy = -r*Y - a*b*X
plt.streamplot(X,Y,dx, dy, density=2, cmap= 'jet', arrowsize=1)

#plot numerical solution:
plt.plot(sol.y.T[:, 0], sol.y.T[:, 1])

#plot your linearized solution:
plt.plot(x(t),y(t))
```

5 Additional Information

The system setup here follows from a process called *nondimensionalization*, where constants are divided through and lumped into new, dimensionless variables. The original system for ENSO follows:

$$\frac{dT_E}{dt} = -cT_E + \gamma'(h_w + b'T_E) - \epsilon'(h_w + b'T_E)^3 \tag{8}$$

$$\frac{dh_w}{dt} = -r'h_w - \alpha' b' T_E \tag{9}$$

Where T_E is the average sea surface temperature in the eastern pacific, and h_w is the thermocline depth anomaly in the western pacific. The thermocline depth describes the boundary between the warmer, top layer of the ocean and the cooler layer below it. The rest of the constants are positive.
Next, we can follow the process of nondimensionalization. There are two reference parameters, T_0 and h_0 - the standard eastern ocean temperature and western thermocline depth.

Variable definitions	
Variable	Value in terms of ODE
x	$\frac{T_E}{T_0}$
y	$\frac{h_w}{h_0}$
τ	ct
r	$\frac{r'}{c}$
α	$\frac{\alpha'}{c}$
γ	$\frac{h_0\gamma'}{T_0 c}$
b	$\frac{T_0}{h_0}b'$
ϵ	$\frac{h_0^3}{T_0 c}\epsilon'$

Generally, $h_0 = 150m$, $T_0 = 7.5K$, and $c = 2$ months. For the equations below, $\dot{x} = \frac{dx}{d\tau}$. This gives us the original system mentioned in (1) and (2):

$$\dot{x} = -x + \gamma(bx + y) - \varepsilon(bx + y)^3 \tag{10}$$

$$\dot{y} = -ry - \alpha bx \tag{11}$$

References

[1] Kaper, H. G, and Hans Engler. *Mathematics and Climate*. Philadelphia, Pennsylvania: Society for Industrial and Applied Mathematics SIAM, 3600 Market Street, Floor 6, Philadelphia, PA 19104, 2013. Print.