Lecture 5: Nonlinear Systems and Stability

Definitions

Nonlinear ODEs are ODEs in which one or more term is nonlinear for the dependent variables. For example, $x_1 x_2$, x^2, $x_1 \frac{dx_1}{dt}$ or $\exp(x)$ are nonlinear terms. Unlike linear systems, nonlinear systems may have completely different outcomes depending on initial conditions. From different initial conditions, the same system may move to one point and stay, may repeat the same cycle forever, or may expand infinitely. Nonlinear systems analysis allows us to predict if and when the solution to a set of differential equations will display these various behaviors.

A nonlinear ODE system with n variables, which we will call x_i, has n functions, $f_i(x, t)$, which determine the derivatives at the state $\vec{x}$ and time t. That is:

$$\frac{d\vec{x}}{dt} = \begin{bmatrix} dx_1(t)/dt \\ \vdots \\ dx_n(t)/dt \end{bmatrix} = \begin{bmatrix} f_1(x_1, \dots x_n, t) \\ \vdots \\ f_n(x_1, \dots x_n, t) \end{bmatrix} = \vec{f}(\vec{x}, t)$$

Thus, $\vec{x}$ indicates the <u>state</u> of the system by identifying the value of each variable in the system. If the systems remains in a state indefinitely, then this state is an <u>equilibrium point</u>, $\vec{x}_{eq}$, of the system. A nonlinear system may have no, one, or multiple equilibrium points. Since systems often have multiple equilibrium points, we refer to the stability of each equilibrium point, rather than to the stability of the entire system, as we did for linear systems.

If an equilibrium point is <u>stable</u>, then any value of $\vec{x}$ sufficiently close to $\vec{x}_{eq}$ will decay to $\vec{x}_{eq}$; that is, any small perturbation from the equilibrium point will decay to zero. Near an <u>unstable</u> equilibrium point, the system will respond to some infinitesimal perturbation by leaving the equilibrium point. A <u>saddle point</u> is a specific type of unstable equilibrium point, in which the solution will approach the equilibrium point from one or more directions, but leave from one or more other directions. Near a <u>marginally or neutrally stable</u> equilibrium point, the system maintains a small perturbation. We refer to equilibrium points as a <u>node</u> if the solution approaches or leaves without oscillations, and a <u>focus</u> if the solution oscillates.

Sometimes solutions repeat the same cycle infinitely. Such a solution will revisit each point on this solution cycle, but since the system always changes, this is not a steady state. If the system is oscillating around a <u>neutrally stable focus</u>, then the oscillations repeat exactly if there is no additional perturbation, but the size of the oscillations depend on the initial conditions, so are affected by past perturbations. However, other systems form a <u>stable limit cycle</u> in which the system converges to the same cyclical solution from a wide range of initial conditions. In this case, there must be an unstable focus or node inside the limit cycle. However, a system with a single unstable node or focus could also simply expand infinitely. Thus, a stable limit cycle is identified from the convergence of multiple randomly chosen initial conditions to the same repeating cycle.

Thus nonlinear systems may have stable solutions such as stable nodes or foci, or stable limit cycles, to which the solutions from many initial conditions converge. We will refer to these as <u>stable solutions</u>. Since different initial conditions can result in the different stable solutions in nonlinear systems, it is useful to characterize this. The set of initial conditions (or system states)

1

that will eventually converge to one particular stable solution is called the <u>basin of attraction</u> for that solution. A <u>separatrix</u> is a line or n-1-dimensional surface that separates two basins of attraction. You can think of the basin of attraction as the watershed that drains into a specific body of water and the separatrix as a continental divide that determines whether a drop of rain will eventually flow to the Atlantic or Pacific oceans.

Equilibrium Points and Stability.

Because the equilibrium point $\vec{x}_{eq}$ of a system is a point where the system is not changing, the derivatives must all be zero. When we set one of the derivatives to zero, we have an algebraic equation: $f_i(x_1, \ldots x_n, t) = 0$. The set of values $\vec{x}$ that satisfy this equation at all times, t, is called a <u>nullcline</u>. We thus find the equilibrium points by simultaneously solving all the nullclines, using either analytic or numeric methods, or both.

We determine the stability of an equilibrium point by linearizing the system about the equilibrium point. This means we find the multidimensional second derivative of the system evaluated at the equilibrium point, which we call the <u>Jacobian.</u>

$$J = \left[\frac{\partial f_i}{\partial x_j}\right] = \begin{bmatrix} \dfrac{\partial f_1}{\partial x_1} & \cdots & \dfrac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \dfrac{\partial f_n}{\partial x_1} & \cdots & \dfrac{\partial f_n}{\partial x_n} \end{bmatrix}$$

As long as this system is close to the equilibrium point, the Jacobian can be used to predict the behavior of the system in the same way that the matrix A does in the state space form of a linear system. Thus, the eigenvalues of the Jacobian give the stability of this particular equilibrium point in a nonlinear system. We call the process of finding equilibrium points and determining their analysis '<u>nonlinear system stability analysis</u>'. By finding equilibrium points and evaluating the Jacobian, we can apply many of the same tools as we learned for linear systems. If we can identify equilibrium points and calculate Jacobians analytically, we have a powerful tool to analyze the system for all parameter values. When this is not possible, we will use numeric tools to analyze the system behavior in the instance of specific parameter values.

Numeric Tools for Nonlinear Systems.

<u>Plot using the Phase Plane method</u>. After solving a time-dependent ODE, you can plot a trajectory by plotting each variable as a function of time. However, it is often useful to plot the system state on a graph with one axis for each variable. This is called a <u>phase plane plot</u>, since it was originally used to graph the value and derivative of the output function, and the derivative is often called the phase. However, there is no requirement that one variable be the derivative of the other, and the term phase plane is used generally to describe a plot of the state variables against each other instead of against time. You can plot the nullclines, equilibrium points, trajectories (or simply derivative values and directions, as arrows), and even eigenvectors on a phase plane plot. To plot individual trajectories is trivial, although you may want to use a different symbol to indicate the start vs the end of the trajectory. To plot the little arrows you sometimes see on phase plane plots, you need to simply calculate the derivative vector (the Jacobian) for each point in an array, and plot this vector starting from the initial value.

Calculating Equilibrium Points. If we cannot obtain an analytic simultaneous solution to the nullclines, we have several alternative approaches. First, it is useful to plot the nullclines on a graph to identify the number and approximate location of the intersections of the nullclines. To plot the nullclines, rewrite each nullcline equation to get one variable as an algebraic function of the others. For a system with two variables, the nullclines are lines in a plane, while the nullclines are planar curves in 3D space for systems of three variables, and higher order systems are difficult to evaluate visually. While plotting nullclines is helpful for visualizing equilibrium points, it does not provide a calculation of the values of each variable at an equilibrium point.

In some cases (like lab this week), you may be able to numerically solve for the intersections of nullclines after some algebraic manipulation, even if you cannot derive an algebraic expression for the equilibrium points. If all else fails, you can use an optimization function such as fminsearch in MATLAB. For this method, you need to minimize a single scalar objective function (type doc fminsearch if you haven't used it before). For example, you can define the objective function to be the sum of the squares of the derivative functions: $obj = \sum_{i=1}^{n} f_i^2(x,t)$. Fminsearch and similar optimizers only finds the local minimum, so any procedure involving optimization needs to be used with multiple initial guesses, each solution needs to be checked for whether it is an equilibrium point, and still is not guaranteed to find all equilibrium points. However, in practice, this method can usually be made to work.

Regardless of how you identify an equilibrium point, you can verify it by plugging the values into the ODE functions (numerically or analytically) to ensure all derivatives are zero, and/or plotting the trajectories using the equilibrium point as the initial conditions to ensure the system remains constant. (However, small numeric errors can accumulate to cause a system to diverge from an unstable equilibrium point over sufficiently long time).

Calculating Stability. We can numerically evaluate the Jacobian matrix at the equilibrium points, and we can use numeric linear algebra to calculate eigenvalues.

Testing effects of initial conditions. It is simple to determine numerically whether any specific point flows into one or another solution, by numerical integration of the ODE. However, it is usually difficult to identify an analytic equation for a separatrix that defines the basins of attraction. Instead, it is often sufficient to illustrate graphically the separatrix and basins of attraction by plotting the trajectories or the derivative arrows on a phase plane plot, as described above. Either method illustrates the system flow.
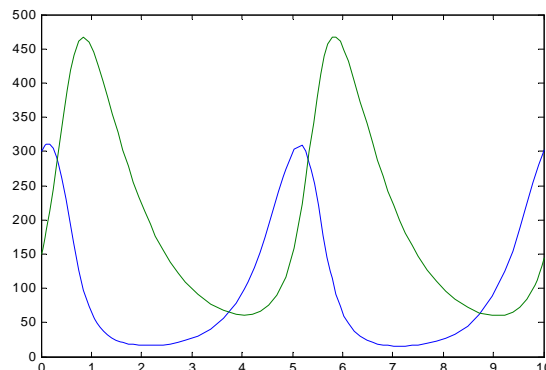
Example 1: Lotka-Volterra Model

The Lotka Volterra predator prey model has been used to model population dynamics in ecological systems. The equations are:

$$\frac{dH(t)}{dt} = rH(t) - aH(t)P(t)$$

$$\frac{dP(t)}{dt} = bH(t)P(t) - mP(t)$$

Where the variable H represents the 'hare', with a birth rate rH, and a death rate aHP. The variable P represents the predators, with a prey-dependent birthrate bHP, and a death rate m. This model disregards death of the prey by any means other than predation, and assumes that there is enough food for the prey.

The solution above illustrates the model behavior with the variables r= 2, a = 0.01, b = 0.01, m = 1, and initial conditions H(0) = 300 (blue), P(0) = 150 (green).

<u>Find the equilibrium points</u> by simultaneous solution of the nullclines: $\vec{f}(\vec{x}, t) = 0$. For our Lotka Volterra model, we need to solve:

$$\frac{dH(t)}{dt} = rH(t) - aH(t)P(t) = 0$$

$$\frac{dP(t)}{dt} = bH(t)P(t) - mP(t) = 0$$

We manipulate this to identify two solutions for each nullcline:

The nullcline for $\frac{dH}{dt} = 0$ is $H(t)(r - aP(t)) = 0$, so that either $H = 0$ or $P = \frac{r}{a} = 200$.

The nullcline for $\frac{dP}{dt} = 0$ is $P(t)(bH(t) - m) = 0$, so either $P = 0$ or $H = \frac{m}{b} = 100$.

In the phase plate plot above, we show the solutions to $\frac{dH}{dt} = 0$ in red, and $\frac{dP}{dt} = 0$ in blue. The equilibrium points are the points where a red and blue line intersect. There are thus two equilibrium points: $(H, P) = (0,0)$ or $(H, P) = \left(\frac{m}{b}, \frac{r}{a}\right)$. We can verify this by solving the ODE with initial conditions $(H, P) = \left(\frac{m}{b}, \frac{r}{a}\right)$, to see if
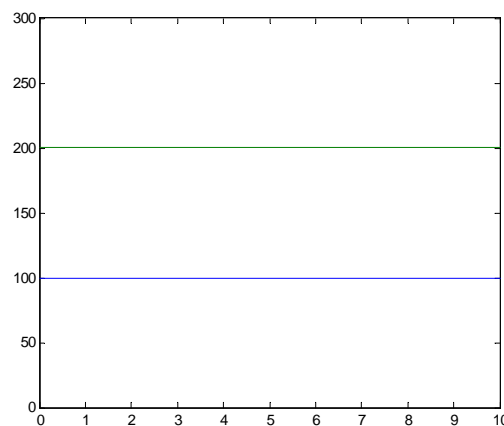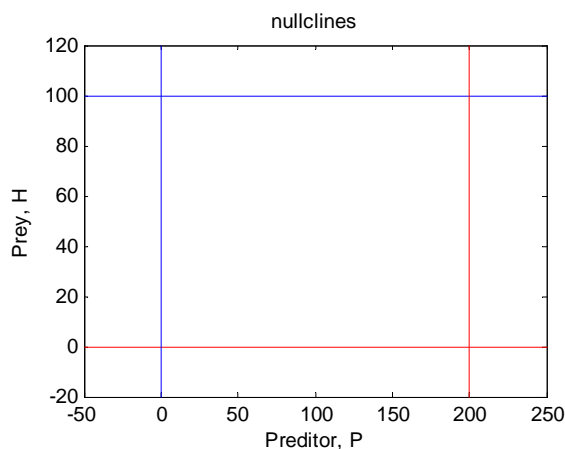
the system remains in the same state, as shown, or by calling the ODE function to return the derivative at this value.

<u>Characterize the Stability</u>.

First, obtain Jacobian analytically, using partial differentiation.

$\frac{\partial f_H}{\partial H} = r - aP$, $\frac{\partial f_H}{\partial P} = -aH$, $\frac{\partial f_P}{\partial H} = bP$, and $\frac{\partial f_P}{\partial P} = bH - m$, so:

$$J = \begin{bmatrix} r - aP & -aH \\ bP & bH - m \end{bmatrix}$$

Second, we evaluate this Jacobian at each equilibrium point:

$$J(0,0) = \begin{bmatrix} r & 0 \\ 0 & -m \end{bmatrix} = \begin{bmatrix} 2 & 0 \\ 0 & -1 \end{bmatrix}$$

$$J\left(\frac{m}{b}, \frac{r}{a}\right) = \begin{bmatrix} r - ar/a & -am/b \\ br/a & bm/b - m \end{bmatrix} = \begin{bmatrix} 0 & -am/b \\ br/a & 0 \end{bmatrix} = \begin{bmatrix} 0 & -1 \\ 2 & 0 \end{bmatrix}$$

Third, we calculate the eigenvalues. We can do this numerically, with:

```
J1 = [r 0;0 -m];
```

```
lambda1 = eig(J1).
```

This returns -1 and +2. This indicates that the (0,0) equilibrium point is a saddle point. If we approach it from one direction, the solution blows up (consider a few Hares, but no Predators, so the hare population expands, since dH/dt > 0, and dP/dt = 0), while if we approach it from another, it decays (consider no Hares, but a few Predators, then the Predators die out, since dH/dt = 0 and dP/dt < 0). If we take a small deviation that has finite values of both, then we see an oscillation that approaches then leaves this equilibrium point without quite hitting it, as shown in the phase plane and time-trajectories below.
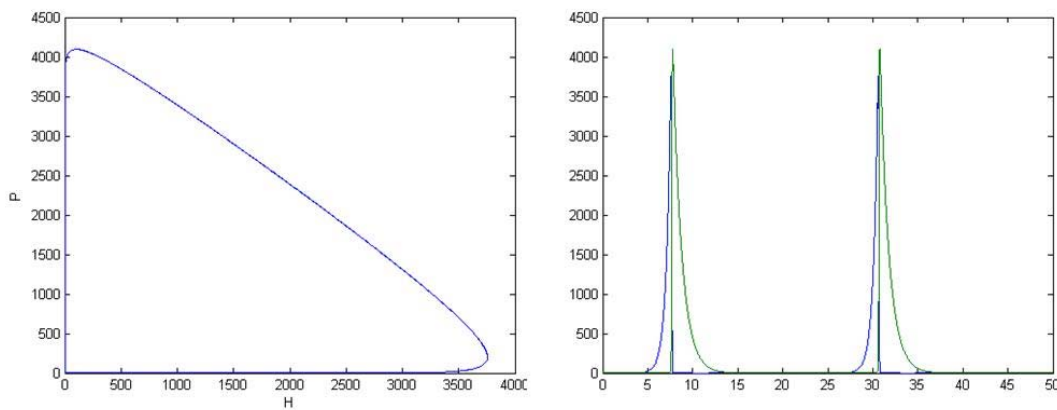
For the other equilibrium point, we calculate: `J2 = [0 -a*m/b; b*r/a 0]; lambda2 = eig(J2),` which returns 0 + 1.4142i and 0-1.4142i. This indicates that this point is a neutrally stable focus. For small deviations, we would expect to oscillate without either approaching or leaving the equilibrium point. We can verify this:
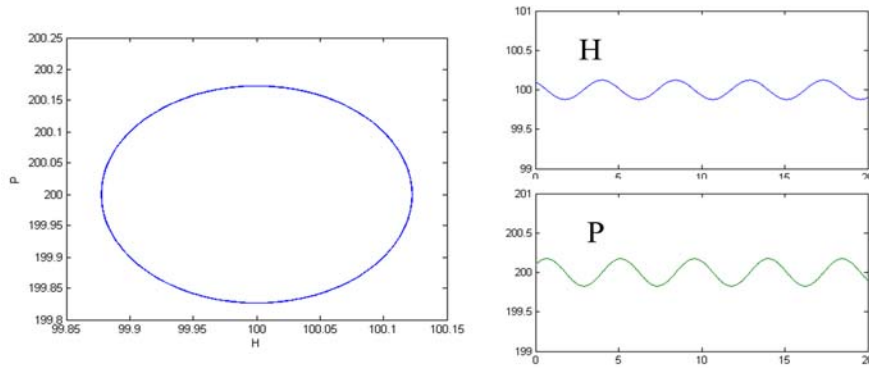
For the Lotka Volterra example, all positive initial conditions oscillate, so there is no separatrix or basins of attraction to identify. However, the size of the oscillations depends on the initial conditions since it is not a stable limit cycle but rather oscillation around a neutrally stable focus. Solutions with exactly zero predator or prey either blow up or collapse to the saddle point, respectively. We don't address solutions with nonphysiolgical initial conditions, such as negative values for this system, so our analysis is complete.
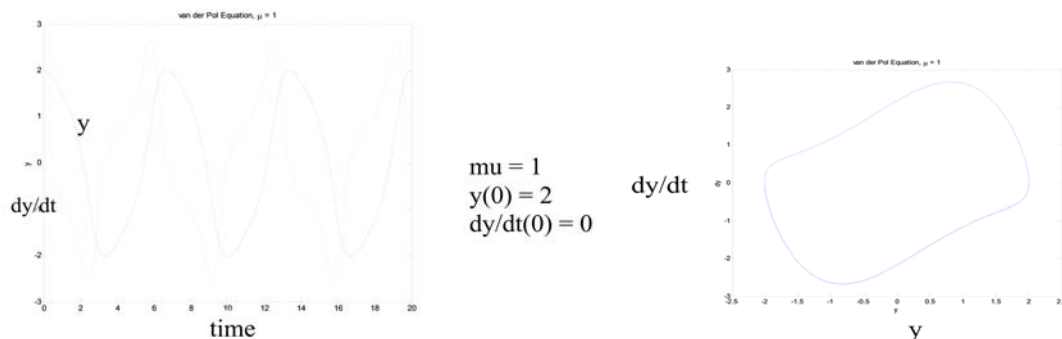
Example 2: Van der Pol.

The Van der Pol equation was proposed in 1928 to model oscillations in the heart. It is a second-order nonlinear differential equation in one variable:

$$\frac{d^2y}{dt^2} - \mu(1-y^2)\frac{dy}{dt} + y = 0$$

To write this as a set of first order ODEs, we let z = dy/dt to get:

$$\frac{dy}{dt} = z$$

$$\frac{dz}{dt} = \mu(1-y^2)z - y$$

This system also oscillates:

For equilibrium, we must have $0 = \frac{dy}{dt} = z$ and $0 = \frac{dz}{dt} = \mu(1-y^2)z - y$. Solving the first yields z = 0, and substituting this into the second equation yields y = 0. Thus, the only equilibrium point occurs at (0,0).

6

The Jacobian matrix is $J = \begin{bmatrix} 0 & 1 \\ -2\mu yz - 1 & \mu(1 - y^2) \end{bmatrix}$. Evaluated at the only equilibrium point, (0,0), this gives $J = \begin{bmatrix} 0 & 1 \\ -1 & \mu \end{bmatrix}$. If $\mu = 1$, then $J = \begin{bmatrix} 0 & 1 \\ -1 & 1 \end{bmatrix}$, which has eigenvalues $\alpha = \frac{(1 \pm \sqrt{-3})}{2}$. Thus, (0,0) is an unstable focus, so the system should oscillate away from this value. Indeed, if we plot the trajectory below starting at (y, dy/dt) = (0.1,0), we see the oscillation away from the origin:
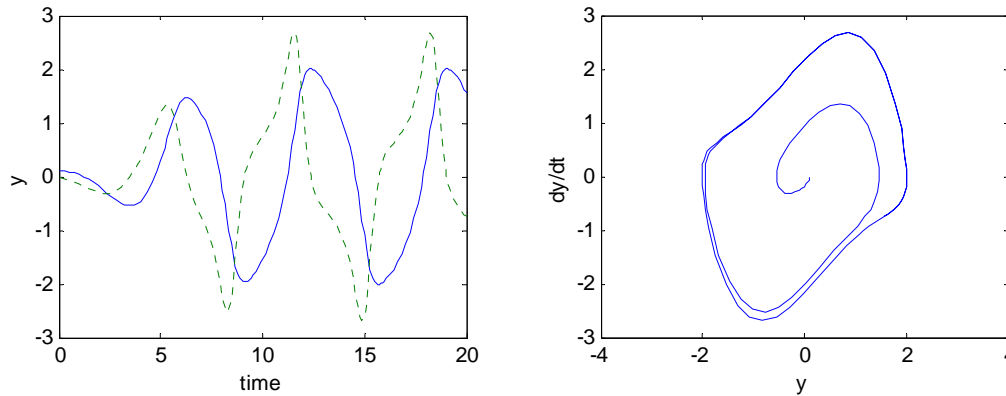
Summary:
1. Calculate the equilibrium points by simultaneous solution of the nullclines, $\vec{f}(\vec{x}, t) = 0$.
    a. If there is an analytic solution, this is most informative.
    b. This can be done graphically for specific parameter values by plotting all the nullclines, at least in a 2D system.
    c. This can be done numerically by minimizing the sum of the squares of the derivatives as the variables change, but the local minimum is not necessarily 0, and thus the system state identified is not necessarily an equilibrium point. Moreover, while one can check local minima after starting with a range of initial conditions, there is no way to know for sure when all equilibrium points have been identified.
    d. If you start a simulation at an equilibrium point, it should not change.
2. We linearize the system at each equilibrium point by calculating the Jacobian, $J = \left[\frac{\partial f_i}{\partial x_j}\right]$. An analytic solution is normally straightforward, but symbolic math software can help.
3. We determine stability of the equilibrium point by finding the eigenvalues of the Jacobian, which give the same stability information as the roots of the characteristic equation for a linear system.
    a. This can be done numerically from the analytic form of the Jacobian, given a specific set of parameter values.
    b. If you start a simulation near an equilibrium point, it should go to or away from that point for a stable and unstable equilibrium point, respectively.