Lab 9: Linear Approximations to Surfaces and Vector Fields

The idea of a linear approximation to a nonlinear object such as a curve has been one of the most powerful in this course. In this lab, you’ll extend this idea to functions of two variables: surfaces and vector fields.

**Tangents to surfaces**

You can plot functions of two variables in Sage using the `plot3d` function. It works just like `plot`, except that you have to specify plotting ranges for both x and y. For example, to plot the function \( f(x, y) = x^2 - xy + y^2 \), you would use the command `plot3d(x^2-x*y+y^2, (x,-4,4), (y,-4,4))`, having first declared y as a symbolic variable.

**Exercise 1.** Plot the function \( f(x, y) = x^2 + y^2 \) and use the interactive viewer to look at the graph from various angles.

We now want to find the tangent plane to \( f(x, y) = x^2 + y^2 \) at the point \((0, 0, 0)\). The key point to keep in mind is that a plane has two slopes – one in the x-direction and one in the y-direction. (The equation for a plane passing through the origin is \( z = k_1 x + k_2 y \), where \( k_1 \) and \( k_2 \) are the two slopes.) We can find these slopes by taking a slice through the graph and plotting the resulting curve.

**Exercise 2.** What does \( f(x, y) = x^2 + y^2 \) become when \( y = 0 \)? Plot the resulting function (in two dimensions). Then, find and plot its tangent at the origin.

**Exercise 3.** Do the same thing for \( x = 0 \). Make sure to label the axes in your two-dimensional graph appropriately.

**Exercise 4.** You now have the slopes of \( f(x, y) \) in both the x and y directions at the origin. Use them to add this tangent plane to the plot of the function. HINT: The equation for the tangent plane will look rather simple.
The general equation for a plane passing through \((0, 0, 0)\) is \(z = k_1x + k_2y\), but what if there’s another point we want to make sure the plane passes through, say \((3, 4, 5)\)? We might say that we want \((3, -4, 5)\) to act as the origin and just shift the axes. In our shifted coordinate system, whose axes we will call \(X\), \(Y\) and \(Z\), the \(X\) coordinate of any point will be 3 units less than its \(x\) coordinate. Similarly, its \(Y\) coordinate will be 4 units more than its \(y\) coordinate, and its \(Z\) coordinate will be 5 units less than its \(z\) coordinate.

Since Sage doesn’t know the new coordinate system, we have to build the conversion into the equation for a plane. We’ll replace \(x\) with \(x - 3\), \(y\) with \(y + 4\) and \(z\) with \(z - 5\). The new equation is therefore \(z - 5 = k_1(x - 3) + k_2(y + 4)\). More generally, the equation for a plane passing through the point \((x_0, y_0, z_0)\) is \(z - z_0 = k_1(x - x_0) + k_2(y - y_0)\) or, in plotting-friendly form, \(z = k_1(x - x_0) + k_2(y - y_0) + z_0\).

**Exercise 5.** Plot the plane passing through the point \((2, 3, -7)\) whose slope in the \(x\)-direction is 2 and whose slope in the \(y\)-direction is -1. To make sure your graph is correct, plot the point using the `point3d` command, whose syntax is the same as that of `point`.

**Exercise 6.** Plot the function \(f(x, y) = x^2 + y^2\) and the plane tangent to it at the point \((1, -1, 2)\). You may want to take advantage of the `opacity` plotting option, which ranges from 0 (completely transparent) to 1 (completely opaque).

**Exercise 7.** Plot another function of \(x\) and \(y\). Then, pick two points on the function and plot the planes tangent to it at those points. You can find the slopes of these planes either graphically as above or using the `diff` command.

**Linearizing differential equations**

Linear approximation (also known as linearization) can help us study the behavior of equilibria of systems of differential equations. Even though the vast majority of differential equations, as well as of biological models that use differential equations, are nonlinear, they behave like linear systems if they are near an equilibrium.

Consider the Holling-Tanner predator-prey model, in which \(x\) is the prey population and \(y\) is the predator population. The equations we will use are \(x' = rx(1 - \frac{x}{10}) - \frac{x}{10}y\) and \(y' = \frac{1}{10}y(1 - \frac{x}{2})\). The parameter \(r\) will vary.
Exercise 8. When \( r = 0.2 \), the Holling-Tanner model has an equilibrium at approximately \((0.0239, 0.24)\). Plot the vector field and mark this point. Then, zoom in on it, viewing the vector field at several different magnifications. What kind of equilibrium is this? HINT: If you can’t tell, add a trajectory to your plot.

The linear approximation to a vector field \( x' = f_1(x, y), y' = f_2(x, y) \) is a matrix of the partial derivatives of each function with respect to each variable. This matrix, \[
\begin{bmatrix}
\frac{\partial f_1}{\partial x} & \frac{\partial f_1}{\partial y} \\
\frac{\partial f_2}{\partial x} & \frac{\partial f_2}{\partial y}
\end{bmatrix}
\]
is called the Jacobian and can be computed in Sage using the command \( \text{jacobian([f1, f2], [x, y])} \). (Of course, you could just use \( \text{diff} \) and make a matrix of the results.)

Exercise 9. Set \( r \) to 0.2 and compute the Jacobian for the Holling-Tanner model. Make sure to assign the result to a variable, as you’ll need it for a future exercise. Use \( \text{show} \) to view your result.

The Jacobian you obtained is a symbolic expression that doesn’t contain particular values for \( x \) and \( y \). In order to get useful output for its eigenvalues, we need to substitute numerical values for these variables. This is done using the following syntax. Suppose you have a symbolic expression called \( \text{sym} \). To substitute the values \( x = 1 \) and \( y = 2 \) into \( \text{sym} \), use \( \text{sym.subs({x:1, y:2})} \). (As you might remember, the object in curly braces is called a dictionary.)

Exercise 10. Substitute the equilibrium values of \( x \) and \( y \) into the Jacobian you found previously. Then, compute the eigenvalues of the resulting matrix. Use Table 1 to classify this equilibrium.

Recall that in one dimension, the equation for a linear approximation is \( \Delta f = \frac{df}{dx} \Delta x \). This is a linear function of \( \Delta x \), not \( x \). If, for example, \( \Delta x = 0.2 \), we can write out the equation as \( \Delta f = \frac{df}{dx} (x - 0.2) \). You can think of this as shifting the horizontal axis.

The same idea applies in multiple dimensions. Therefore, when plotting linear approximations to vector fields, you have to shift the axes appropriately, just like you did when plotting tangents to surfaces.
**Exercise 11.** Plot the vector field corresponding to the numerical Jacobian you found in the previous exercise. How does it compare to what you see near the equilibrium of the nonlinear system? HINT: You can’t plot a vector field using a matrix. Just write out the equations the matrix represents and use them.

**Exercise 12.** When $r = 0.5$, the equilibrium values of $x$ and $y$ are 0.796 for both variables. Repeat Exercises 8-11 for these values.