3.1: Introduction to Systems of ODEs

Often we do not have just one dependent variable and just one differential equation, we may end up with systems of several equations and several dependent variables even if we start with a single equation. If we have several dependent variables, suppose \( y_1, y_2, \ldots, y_n \), then we can have a differential equation involving all of them and their derivatives. For example, \( y''_1 = f(y'_1, y'_2, y_1, y_2, x) \). Usually, when we have two dependent variables we have two equations such as

\[
\begin{align*}
  y''_1 &= f_1(y'_1, y'_2, y_1, y_2, x) \\
  y''_2 &= f_2(y'_1, y'_2, y_1, y_2, x)
\end{align*}
\]

for some functions \( f_1 \) and \( f_2 \). We call the above a system of differential equations. More precisely, the above is a second order system of ODEs.

Example (PageIndex{1}): 

Sometimes a system is easy to solve by solving for one variable and then for the second variable. Take the first order system

\[
\begin{align*}
  y'_1 &= y_1, \\
  y'_2 &= y_1 - y_2
\end{align*}
\]

with initial conditions of the form \( y_1(0) = 1 \) and \( y_2(0) = 2 \).

We note that \( y_1 = C_1 e^x \) is the general solution of the first equation. We then plug this \( y_1 \) into the second equation and get the equation \( y'_2 = C_1 e^x - y_2 \), which is a linear first order equation that is easily solved for \( y_2 \). By the method of integrating factor we obtain

\[
\begin{align*}
  e^x y_2 &= \frac{C_1}{2} e^{2x} + C_2 \\
  y_2 &= \frac{C_1}{2} e^2 + C_2 e^{-x}
\end{align*}
\]
The general solution to the system is, therefore,

\[ y_1 = C_1e^x, \] and \[ y_2 = \frac{C_1}{2}e^x + C_2e^{-x}. \]

We now solve for \((C_1, C_2)\) given the initial conditions. We substitute \(x=0\) and find that \(C_1 = 1\) and \(C_2 = \frac{3}{2}\). Thus the solution is:

\[ y_1 = e^x, \] and \[ y_2 = \frac{1}{2}e^x + \frac{3}{2}e^{-x}. \]

Generally, we will not be so lucky to be able to solve for each variable separately as in the example above, and we will have to solve for all variables at once.

As an example application, let us think of mass and spring systems again. Suppose we have one spring with constant \((k)\), but two masses \((m_1)\) and \((m_2)\). We can think of the masses as carts, and we will suppose that they ride along a straight track with no friction. Let \((x_1)\) be the displacement of the first cart and \((x_2)\) be the displacement of the second cart. That is, we put the two carts somewhere with no tension on the spring, and we mark the position of the first and second cart and call those the zero positions. Then \((x_1)\) measures how far the first cart is from its zero position, and \((x_2)\) measures how far the second cart is from its zero position. The force exerted by the spring on the first cart is \( k(x_2 - x_1) \), since \(x_2 - x_1\) is how far the spring is stretched (or compressed) from the rest position. The force exerted on the second cart is the opposite, thus the same thing with a negative sign.

Newton’s second law states that force equals mass times acceleration. So the system of equations governing the setup is

\[ m_1 x''_1 = k(x_2 - x_1) \]
\[ m_2 x''_2 = -k(x_2 - x_1) \]

In this system we cannot solve for the \((x_1)\) or \((x_2)\) variable separately. That we must solve for both \((x_1)\) and \((x_2)\) at once is intuitively clear, since where the first cart goes depends exactly on where the second cart goes and vice-versa.

Before we talk about how to handle systems, let us note that in some sense we need only consider first order systems. Let us take an \((n^{th})\) order differential equation

\[ y^{(n)} = F(y^{(n-1)}, ..., y', y, x). \]

We define new variables \((u_1, ..., u_n)\) and write the system

\[ u'_1 = u_2 \]
\[ u'_{n-1} = u_n \]
\[ u'_n = F(u_n, u_{n-1}, \ldots, u_2, u_1, x) \]

We solve this system for \( u_1, u_2, \ldots, u_n \). Once we have solved for the \( u \)'s, we can discard \( u_2 \) through \( u_n \) and let \( y = u_1 \). We note that this \( y \) solves the original equation.

A similar process can be followed for a system of higher order differential equations. For example, a system of \( k \) differential equations in \( k \) unknowns, all of order \( n \), can be transformed into a first order system of \( n \times k \) equations and \( n \times k \) unknowns.

Example \( \PageIndex{2} \):

Sometimes we can use this idea in reverse as well. Let us take the system

\[ x' = 2y - x \]
\[ y' = x \]

where the independent variable is \( t \). We wish to solve for the initial conditions \( x(0) = 1 \) and \( y(0) = 0 \).

If we differentiate the second equation we get \( y'' = x' \). We know what \( x' \) is in terms of \( x \) and \( y \), and we know that \( x = y' \).

\[ y'' = x' = 2y - x = 2y - y. \]

We now have the equation \( y'' + y' - 2y = 0 \). We know how to solve this equation and we find that \( y = C_1 e^{-2t} + C_2 e^t \). Once we have \( y \) we use the equation \( y' = x \) to get \( x \).

\[ x = y' = -2C_1 e^{-2t} + C_2 e^t \]

We solve for the initial conditions \( 1 = x(0) = -2C_1 + C_2 \) and \( 0 = y(0) = C_1 + C_2 \). Hence, \( C_1 = -C_2 \) and \( 1 = 3C_2 \). So \( C_1 = \frac{1}{3} \) and \( C_2 = \frac{1}{3} \). Our solution is

\[ x = \frac{2e^{-2t} + e^t}{3}, \quad y = \frac{-e^{-2t} + e^t}{3} \]

Exercise \( \PageIndex{1} \):

UC Davis ChemWiki is licensed under a Creative Commons Attribution-Noncommercial-Share Alike 3.0 United States License.
Plug in and confirm that this really is the solution.

It is useful to go back and forth between systems and higher order equations for other reasons. For example, the ODE approximation methods are generally only given as solutions for first order systems. It is not very hard to adapt the code for the Euler method for first order equations to handle first order systems. We essentially just treat the dependent variable not as a number but as a vector. In many mathematical computer languages there is almost no distinction in syntax.

- In fact, this is what IODE was doing when you had it solve a second order equation numerically in the IODE Project III if you have done that project.

The above example is what we call a linear first order system, as none of the dependent variables appear in any functions or with any higher powers than one. It is also autonomous as the equations do not depend on the independent variable \(t\). For autonomous systems we can easily draw the so-called direction field or vector field. That is, a plot similar to a slope field, but instead of giving a slope at each point, we give a direction (and a magnitude).

The previous example \(( x' = 2y - x, y' = x)\) says that at the point \((x,y)\) the direction in which we should travel to satisfy the equations should be the direction of the vector \((2y - x, x)\) with the speed equal to the magnitude of this vector. So we draw the vector \((2y-x,x)\) based at the point \((x,y)\) and we do this for many points on the \((xy)\) -plane. We may want to scale down the size of our vectors to fit many of them on the same direction field (Figure 3.1).

We can now draw a path of the solution in the plane. That is, suppose the solution is given by \((x = f(t), y = g(t))\) then we can pick an interval of \((t)\) (say \((0 \geq t \geq 2)\) for our example) and plot all the points \((( f(t), g(t))\) for \((t)\) in the selected range. The resulting picture is usually called the phase portrait (or phase plane portrait). The particular curve obtained we call the trajectory or solution curve. An example plot is given in Figure 3.2. In this figure the line starts at \((1, 0)\) and travels along the vector field for a distance of 2 units of \((t)\). Since we solved this system precisely we can compute \((x(2))\) and \((y(2))\). We get that \((x(2) \approx 2.475)\) and \((y(2) \approx 2.457)\). This point corresponds to the top right end of the plotted solution curve in the figure.

![Figure 3.1: The direction field for \(( x' = 2y - x, y' = x)\).](image)
Figure 3.2: The direction field for \( x' = 2y - x, \ y' = x \) with the trajectory of the solution starting at \( (1, 0) \) for \( 0 \leq t \leq 2 \).

Notice the similarity to the diagrams we drew for autonomous systems in one dimension. But now note how much more complicated things become if we allow just one more dimension.

Also note that we can draw phase portraits and trajectories in the \( xy \)-plane even if the system is not autonomous. In this case however we cannot draw the direction field, since the field changes as \( t(t) \) changes. For each \( t(t) \) we would get a different direction field.

---

**Contributors**

- Jiří Lebl (Oklahoma State University). These pages were supported by NSF grants DMS-0900885 and DMS-1362337.