MATH 319, WEEK 4: Applications and Numerical Methods

1 Applications - In-Flow / Out-Flow Models

One popular application of differential equations (and in particular, first-order linear differential equations) is in modeling the amount (or concentration) of a substance in a well-stirred tank/vessel subject to constant in-flow and out-flow. Common simple applications are:

- an industrial mixing tank with an entry pipe (pumping the chemical of interest in) and an exit pipe;
- a lake with a inflow (say, a river) feeding a pollutant from upstream and an outflow (also, a river) flowing downstream;
- a tub or sink with a steady inflow (say, a faucet) and a steady outflow (say, a drain).

In all cases, the basic question is the same: If we know the in-flow, and the out-flow, can we determine what actually happens *inside* the tank/lake/tub/etc.?

To answer this question, we must translate this description from words into math. At the most basic level, we believe that

$$\text{[rate of change]} = \text{[rate in]} - \text{[rate out]}.$$

That is to say, at each instance in time, we believe that the rate of change of the overall *amount* of the quantity of interest to equal the amount that is flowing in minus the amount that is flowing out. The question of characterizing the dynamics is therefore only a matter of characterizing the in-flow and the out-flow! Our knowledge of differential equations should handle the rest.

To characterize the in-flow rate, we need a few pieces of information. Firstly, we are likely to be given the overall mixture flow rate in, as well as the concentration of the quantity of interest within that in-flowing mixture. For example, we might know the amount of water which flows into a lake every day, or every week, and we might know the concentration of a particular pollutant within that volume of water. The rate of the amount of the pollutant flowing in is therefore

$$[\text{rate in}] = [\text{volume in}] \times [\text{concentration}]$$

since

$$[\text{volume in}] \times [\text{concentration in}] = \left[\frac{\text{volume}}{\text{time}}\right] \times \left[\frac{\text{amount}}{\text{volume}}\right] = \left[\frac{\text{amount}}{\text{time}}\right]$$

The out-flow is slightly different. Since we are assuming (for simplicity!) that the tank/lake/tub is well-mixed, we may assume that the concentration of the quantity of interest is the same *everywhere* in the tank/lake/tub. In particular, wherever the outflow is located, and however quickly it is removing mixture from the tank/lake/tube, we have

$$[\text{rate out}] = [\text{concentration}] \times [\text{volume out}] = \frac{[\text{amount}]}{[\text{volume}]} \times [\text{volume out}]$$

since

$$\frac{[\text{amount}]}{[\text{volume}]} \times [\text{volume out}] = \left[\frac{\text{amount}}{\text{volume}}\right] \times \left[\frac{\text{volume}}{\text{time}}\right] = \left[\frac{\text{amount}}{\text{time}}\right].$$

The key difference here is that the amount in the above derivation is the *current* amount of the quantity of interest. In other words, it is the unknown function/variable we are trying to model! Another wrinkle is that the volume is the *current* volume of the tank. If the volume of the in-flow and the volume of the out-flow do not balance, the volume of the tank may not be fixed and may in fact be a function of time (imagine filling a bathtub, or emptying a mixing tank).

Example 1: Suppose that there is a factory built upstream of Lake Mendota (volume 0.5 km^3) which introduces a new pollutant to a stream which pumps 1 km^3 of water into the lake every year. Suppose that the net outflow from the lake is also 1 km^3 per year and that the concentration of the pollutant in the inflow stream is 200 kg/km^3. Set up an initial value problem for the amount of pollutant in the lake and solve it. Assuming there is initially no pollutant in the lake, how much pollutant is there are one month? What is the limiting pollutant level?

Solution: We need to set up the model in the form [rate of change]=[rate in]−[rate out]. If we let A denote the amount of the pollutant (in kg), we have

$$[\text{rate of change}] = \frac{dA}{dt}.$$

In order to determine the rate in, we notice that the amount (in kg) coming from the inflow can be given by

$$\begin{aligned}[\text{rate in}] &= [\text{volume rate in}] \times [\text{concentration in}] \\ &= (1 \text{ km}^3/\text{year})(200 \text{ kg/km}^3) = 200 \text{ kg/year}.\end{aligned}$$

The rate out is given by

$$\begin{aligned}[\text{rate out}] &= [\text{volume rate out}] \times [\text{concentration out}] \\ &= (1 \text{ km}^3/\text{year}) \left(\frac{A}{0.5} \text{ kg/km}^3 \right) = 2A \text{ kg/year}.\end{aligned}$$

We can see the units have worked as desired. We can drop them and just focus on the initial value problem

$$\frac{dA}{dt} = 200 - 2A, \quad A(0) = A_0.$$

This is a first-order linear differential equation which in standard form is given by

$$\frac{dA}{dt} + 2A = 200.$$

We can see that we have $p(x) = 2$ and $q(x) = 200$. The necessary integration factor is

$$\rho(t) = e^{\int 2 \, dt} = e^{2t}$$

so that we have

$$\begin{aligned} & e^{2t} \frac{dA}{dt} + 2e^{2t} A = 200e^{2t} \\ \implies \quad & \frac{d}{dt} \left[e^{2t} A \right] = 100e^{2t} \\ \implies \quad & e^{2t} A = 100e^{2t} + C \\ \implies \quad & A(t) = 100 + Ce^{-2t}. \end{aligned}$$

In order to solve for C, we use $A(0) = A_0$ to get

$$A(0) = A_0 = 100 + C \quad \implies \quad C = A_0 - 100.$$

This gives the solution

$$A(t) = 100 + (A_0 - 100)e^{-2t}.$$

For this form, we can easily answer the stated questions. Given an initial pollutant level of zero (i.e. $A_0 = 0$), we have

$$A(t) = 100 - 100e^{-2t}.$$

After one month has passed, we have $t = 1/12$ so that the amount of pollutant is given by

$$A(1/12) = 100 - 100e^{-2(1/12)} \approx 15.3528 \text{ kg}.$$

We can also easily determine the limiting pollutant level by evaluating

$$\lim_{t \to \infty} A(t) = \lim_{t \to \infty} \left[100 + (A_0 - 100)e^{-2t} \right] = 100.$$

In other words, no matter what the initial amount is in the lake, we will always converge toward 100 kg of pollutant distributed throughout the lake. (This should make some sense. We imagine that the limiting level is going to be when the rate in and the rate out are balanced. That occurs for this model when $200 = 2A$ which implies $A = 100$.)

Example 2: Consider a 50 gallon tank which is initial filled with 20 gallons of brine (salt/water mixture) with a concentration of 1/4 lbs/gallon of salt. Suppose that there is an inflow tube which infuses 3 gallons of brine into the tank per minute with a concentration of 1 lbs/gallon. Suppose that there is an outflow tube which flows at a rate of 2 gallons per minute. Set up and solve a differential equation for the amount of salt in the tank. How much salt is in the tank when the tank is full?

Solution: This is slightly different than the previous example because the volume of mixture in the tank *changes* because the inflow and outflow volume rates are different. There is more mixture flowing into the tank than flowing out. Nevertheless, we can incorporate this into our model by noting that the volume of the tank at time t can be given by

$$V(t) = 20 + (3 - 2)t = 20 + t.$$

We can now complete the model as before. We have

$$\frac{dA}{dt} = (3)(1) - (2)\frac{A}{20 + t} = 3 - \frac{2A}{20 + t}, \quad A(0) = 20(1/4) = 5.$$

Again, this is a first-order linear differential equation. We can solve it by rewriting

$$\frac{dA}{dt} + \left(\frac{2}{20 + t} \right) A = 3$$

and determining the integrating factor

$$\rho(t) = e^{\int 2/(20+t)\, dt} = e^{2\ln(20+t)} = (20+t)^2.$$

This gives

$$\begin{aligned}
& (20+t)^2 \frac{dA}{dt} + 2(20+t)A = 3(20+t)^2 \\
\implies \quad & \frac{d}{dt}\left[(20+t)^2 A\right] = 3(20+t)^2 \\
\implies \quad & (20+t)^2 A = (20+t)^3 + C \\
\implies \quad & A(t) = (20+t) + \frac{C}{(20+t)^2}.
\end{aligned}$$

Using the initial condition $A(0) = 5$, we have

$$A(0) = 5 = 20 + \frac{C}{400} \quad \implies \quad C = -6000$$

so that the particular solution is

$$A(t) = (20+t) - \frac{6000}{(20+t)^2}.$$

To answer the question of how much salt will be in the tank when the tank is full, we notice that the tank will be full when $V(t) = 20 + t = 50$, which implies $t = 30$ (i.e. it will take thirty minutes). This gives

$$A(30) = (20+30) - \frac{6000}{(20+30)^2} = 50 - \frac{6000}{2500} = 47.6.$$

It follows that there will be 47.6 lbs of salt in the tank when it is full.

2 Numerical Methods

We may feel pretty optimistic regarding our abilities to solve first-order differential equations at this point, but we have generally been operating so far under the assumptions that (a) solutions exist; and (b) if they exist, we can find them.

Now re-consider our earlier example $\frac{dy}{dx} = x^2 + y^2$. Our toolbox of differential equation solving methods is pretty small so far, but it is growing. As we go through the tools we have accumulated so far for this example, however, we quickly find ourselves frustrated. This differential equation is not directly integrable, it is not separable or first-order linear, it is not (power) homogeneous or Bernoulli, and there is not integrating factor to make it exact. Nothing we have learned so far will help us.

We should not be surprised to learn that there are first-order differential equations which cannot be solved by the elementary methods we have developed so far. In fact, *most* differential equations used in the applied sciences do not have solutions which can be represented in terms of elementary functions (e.g. x^n, $\sin(x)$, $\cos(x)$, e^x, $\ln(x)$, etc.). The differential equation considered above, for instance, only has solutions which can be represented in terms of *Bessel functions* (i.e. this class of functions can only be represented as an infinite series of (potentially non-integer) powers of x!).

Our interest in differential equations does not stop when we fail to be able to solve them, however. Our existence theorem guarantees that solutions exist through every point (x, y) where $f(x, y)$ is continuous, which is *everywhere* for this differential equation. In other words, we know a solution exists! We need to find a way to characterize this solution given that we cannot analytically solve the differential equation.

This seems like an insurmountable task at first glance, but reconsider the *slope field* diagram idea from a few weeks ago. Our intuition then was that the value of $f(x, y)$ at (x, y) corresponded to the slope of the particular solution $y(x)$ through the point (x, y) at the point (x, y). If we graphed a representative sample of slopes (drawn as short lines) in the (x, y)-plane, we could get a good sense of what solutions must look like. We were able to correspond the analytic solutions for several examples to their slope field diagrams.

We notice at this point that, even though we cannot (easily) find the solution $y(x)$ of $\dfrac{dy}{dx} = x^2 + y^2$, it is still relatively easy to construct a direction field diagram. We could create a table of values for $f(x, y)$, or just notice that $f(x, y) \geq 0$ and the steepness of the slope lines grows as we travel along circles radiating out from $(0, 0)$. (That is to say, we have a curve of points with the same slope along the circles $x^2 + y^2 = C$.) If we are careful, we eventually arrive at the direction field picture given in Figure 1(a).

Even though we do not have access to an analytic solution for this differential equation, we can get some sense of what any solution *must* look like. All we have to do (essentially) is connect the lines! It does not take long to come up with a picture that looks something like Figure 1(b).

This process is good for visualization, but it is not rigorous. For instance, consider asking a question like: given the initial value $y(0) = 0$, what is the value of the solution through this point at $x = 1$? We would certainly look

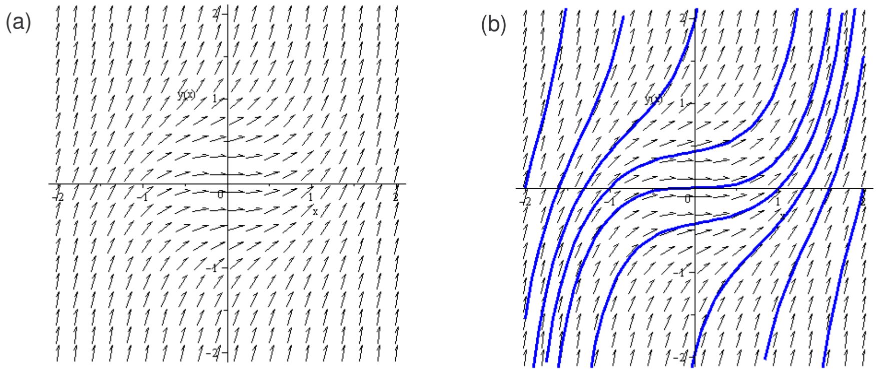

Figure 1: (a) Slope field diagram of $\dfrac{dy}{dx} = x^2 + y^2$. (b) Slope field diagram with solutions.

at our slope field diagram, find the solution through $(0, 0)$ and guess where that curve is going to be when $x = 1$, but we would like to do better.

To consider how we might approach this problem, let's consider the slope field diagram in more depth. We have the following intuition:

1. The slope at a point (x, y) agrees *locally* with the trajectory through the point.

2. A trajectory agrees with the slopes of the arrows at every point it passes through.

This leads us to the following intuition: If we start at a given point (x_0, y_0), *locally* the solution through that point agrees with the solution along the line given by the slope of the arrow. Imagine moving straight along the line at slope $f(x_0, y_0)$ by a small increment in Δx. This gives us a new point (x_1, y_1). At this point, the value of $f(x, y)$ has changed, but so long as the initial increment was small we imagine it has not changed much. So let's continue this process! If we take small increments in x (say $0 < \Delta x \ll 1$) we imagine each step forward in the state space is not far away from the analytic trajectory corresponding to the same initial condition (see Figure 2).

This method is called the forward *Euler's method* and is given explicitly by the formula

$$y_{n+1} = y_n + f(x_n, y_n)\Delta x. \tag{1}$$

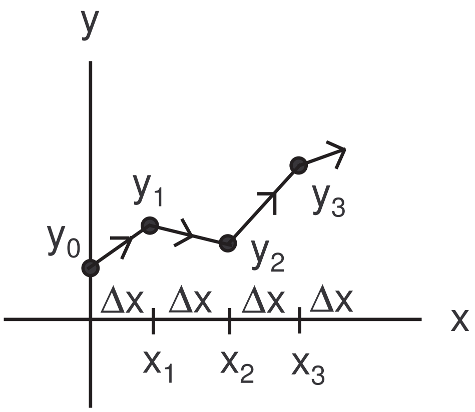

Figure 2: The forward Euler method traces out a solution by jumping forward in increments of Δx along arrows of the slope field diagram.

This formula corresponds exactly to the intuition was offered above. At a point (x_n, y_x), we compute the next state (x_{n+1}, y_{n+1}) by updating the current point by the slope of the vector field at that point $(f(x_n, y_n))$ over a small increment (Δx). We then repeat the process. This is a form of *numerical approximation.*

For our example, we have the update scheme $y_{n+1} = y_n + f(x_n, y_n)\Delta x = y_n + (x_n^2 + y_n^2)\Delta x$ and $x_{n+1} = x_n + \Delta x$. Choosing $\Delta x = 0.1$ and $(x_0, y_0) = (0, 0)$, we have

$$y_1 = y_0 + (x_0^2 + y_0^2)\Delta x = (0) + ((0)^2 + (0)^2)(0.1) = 0.$$

We also have that $x_1 = x_0 + \Delta x = (0) + (0.1) = 0.1$ so that $(x_1, y_1) = (0.1, 0)$. Applying the procedure again, we have

$$y_2 = y_1 + (x_1^2 + y_0^2)\Delta x = (0) + ((0.1)^2 + (0)^2)(0.1) = 0.001.$$

It follows that $(x_2, y_2) = (0.2, 0.001)$. Continuing this procedure, we arrive at the following table of values given in Table 1.

These values represent a *numerical solution.* They are the analogue of plugging specific x values into our solution form $y(x)$. Of course, for this example, we do not have a solution form $y(x)$ so this is as good as we can do.

n	x_n	y_n
0	0	0
1	0.1	0
2	0.2	0.001
3	0.3	0.0050001
4	0.4	0.0140026
5	0.5	0.030022207
6	0.6	0.05511234
7	0.7	0.091416077
8	0.8	0.141251767
9	0.9	0.207246973
10	1.0	0.292542104

Table 1: Table of values for the numerical solution of $dy/dx = x^2 + y^2$ starting at $(0, 0)$ using Euler's method and the step size $\Delta x = 0.1$.

There are several very important notes worth making about this procedure:

1. Beyond a few iterations, this is not a process we want to do by hand. Computers are a necessity, and they are very good (and getting better and better!) at efficiently computing numerical solutions. As computers have become more wide-spread (last fifty years), the emphasis in *applied mathematics* has shifted significantly toward numerical integration, to the point where it is currently probably the most significant approach taken in the field.

2. Numerical integration has two significant drawbacks when it comes to model analysis: (1) It requires a specified initial condition, and (2) it requires specified parameter values. In other words, it can suggest whether a model permits certain behavior (e.g. growth/decay/stability, oscillations, etc.) but can only do so for *one* particular solution at a time. Analytic solutions, if they can be found, are more insightful because they can consider all of this information at the same time.

3. Each step in the process has a error associated to it, so how do we know the numerical solution approximates the actual solution after hundreds or thousands of iterations? Even if each step has a small error, how do we guarantee the cumulation of these errors is small? We will not investigate these concerns in too much detail, but we will make the following notes about ways to increase accuracy:

(a) Choose a smaller time step Δx.

(b) Choose a better numerical scheme (forward Euler is excellent for an accessible introduction to the topic, but *terrible* for bounding the accumulation of errors).

which justifies the form (3).

In order to consider how the forward Euler method performs, consider the following.

Example: With the help of a computer, use the Forward Euler method with step sizes $\Delta x = 0.5, 0.1, 0.01$, and 0.001 to estimate the value of $y(1.5)$ for the differential equation

$$\frac{dy}{dx} = x^2 + y^2, \qquad y(0) = 0. \tag{2}$$

Comment on how these results compare to the "true" value of $y(1.5) = 1.517447537$.

Solution: To re-iterate, the formula for the Forward Euler method is

$$\text{Forward Euler:} \qquad y_{n+1} = y_n + f(x_n, y_n)\Delta x. \tag{3}$$

Notice first of all that, when we specify a particular point in the future we are interested in, we can determine the number of steps required to get there. For instance, we will need $(x_{final} - x_{initial})/\Delta x = 1.5/0.001 = 1500$ computations to produce the estimate for $y(1.5)$ using $\Delta x = 0.001$—so we have our work cut out for us! To do this by hand is infeasible; fortunately, computers can implement such recursive algorithms as Euler's method (and other numerical schemes) very, very quickly.

We can carry out the procedure outlined in the lecture notes by hand to get the first few estimates with our calculators, but for small step-sizes we will definitely have to use a computer. The output from this gives the result contained in the table above.

Δx	$y(1.5)$	error	steps
0.5	0.6328125	0.884635	3
0.1	0.9307268557	0.5867207	15
0.01	1.479113716	0.0383338	150
0.001	1.513502037	0.0039455	1500

We can see that there is a marked improvement by reducing the step size. This makes sense—we have less chance of floating far away from the true trajectory if we take smaller steps before correcting ourselves. We should, however, we disappointed in the tremendous number of steps associated with the refine estimate. Fifteen-hundred steps represents a significant computational expenditure for a few decimal places of accuracy, and it would be reasonable at this point to wonder if there are alternative schemes by which to construct numerical solutions.

2.1 Runge-Kutta Method

The Runge-Kutta method is a popular choice and is known to produce less error per step than the forward Euler, but at the cost of being more computationally intensive during each step. The formulas are given by the following:

$$\text{Runge-Kutta:}\qquad \begin{aligned} y_{n+1} &= y_n + \frac{\Delta x}{6}\left(k_1 + 2k_2 + 2k_3 + k_4\right) \\ &\text{where} \\ k_1 &= f(x_n, y_n) \\ k_2 &= f(x_n + \frac{1}{2}\Delta x, y_n + \frac{1}{2}k_1\Delta x) \\ k_3 &= f(x_n + \frac{1}{2}\Delta x, y_n + \frac{1}{2}k_2\Delta x) \\ k_4 &= f(x_n + \Delta x, y_n + k_3\Delta x). \end{aligned} \tag{4}$$

These equations look crazy at first glance! But before we throw our hands up in defeat, let's see how it performs for the previous example taking $\Delta x = 0.5, 0.1$ and 0.01.

The computations are easy enough to perform for Δx that we will do one step by hand. Thereafter, we will have to rely on a computer—or dedicate a significantly greater amount of time to this course than any of us currently have. As always, we have $x_1 = x_0 + \Delta x = (0) + (0.5) = 0.5$. To compute y_1, we need k_1, k_2, k_3, and k_4. We have

$$k_1 = f(x_0, y_0)\Delta x = x_0^2 + y_0^2 = (0)^2 + (0)^2 = 0$$

and

$$\begin{aligned}
k_2 &= f(x_0 + \frac{1}{2}\Delta x, y_0 + \frac{1}{2}k_1\Delta x) \\
&= (x_0 + \frac{1}{2}\Delta x)^2 + (y_1 + \frac{1}{2}k_1\Delta x)^2 \\
&= ((0) + \frac{1}{2}(0.5))^2 + ((0) + \frac{1}{2}(0)(0.5))^2 = 0.0625
\end{aligned}$$

and

$$\begin{aligned}
k_3 &= f(x_0 + \frac{1}{2}\Delta x, y_0 + \frac{1}{2}k_2\Delta x) \\
&= (x_0 + \frac{1}{2}\Delta x)^2 + (y_0 + \frac{1}{2}k_2\Delta x)^2 \\
&= ((0) + \frac{1}{2}(0.5))^2 + ((0) + \frac{1}{2}(0.0625)(0.5))^2 \\
&= 0.06274414
\end{aligned}$$

and

$$\begin{aligned}
k_4 &= f(x_0 + \Delta x, y_0 + k_3\Delta x) \\
&= (x_0 + \Delta x)^2 + (y_0 + k_3\Delta x)^2 \\
&= ((0) + (0.5))^2 + ((0) + (0.06274414)(0.5))^2 \\
&= 0.250984206.
\end{aligned}$$

That was a lot of work, and we haven't even computed the estimate y_1 yet! We finally have

$$\begin{aligned}
y_1 &= y_0 + \frac{\Delta x}{6}\left(k_1 + 2k_2 + 2k_3 + k_4\right) \\
&= (0) + \frac{0.5}{6}\left((0) + 2(0.0625) + 2(0.06274414) + (0.250984206)\right) \\
&= 0.041789373.
\end{aligned}$$

At this point, we are probably about to throw our hands up and swear off the Runge-Kutta method once and for all. This was a pile of work just to do one time-step! Before we despair too much, however, we should recognize that all the work have done is easily programmed into a computer, and that is exactly what is done in application. Letting my laptop do the rest of the work, in a fraction of a second we have the following estimates:

Δx	$y(1.5)$	error	steps
0.5	1.521061677	0.00361414	3
0.1	1.517473413	0.000025876	15
0.01	1.517447548	0.000000011	150

The reason we have gone through all of this trouble—or rather, let our computers go through all this trouble—should now be clear. The Runge-Kutta method gives a *significantly* better estimate of the true value per step. No matter how ridiculous we find the amount of computation necessary in each step to be, we cannot escape the *overall* efficiency. We have obtained a better estimate of $y(1.5)$ in three steps of the Runge-Kutta method (error= 0.00361414) than we obtained in 1500 iterations of the forward Euler method (error= 0.0039455). It should come as no surprise, therefore, to learn that the forward Euler method—while illustrative and intuitive—is never, ever, ever use in practice. Even though each step is easy to compute, the overall burden of cumulative errors makes it tremendously inefficient.