MATH 320, WEEK 5:
Numerical Approximation

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 consider the example $\dfrac{dy}{dx} = x^2 + y^2$ (Example 5, page 120 of text). 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*. (Bessel functions will not be covered in this course, but to get a sense of how *non-elementary* the required functions can get, 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 slope 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 slope field picture given in Figure 1(a).

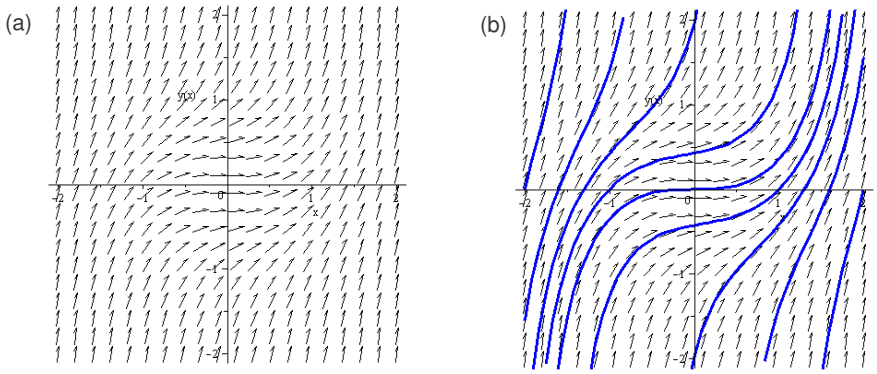

Figure 1: (a) Slope field diagram of $\dfrac{dy}{dx} = x^2 + y^2$. (b) Slope field diagram with solutions.

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 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).

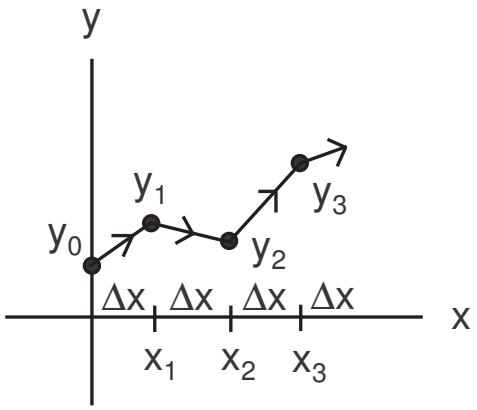

Figure 2: The forward Euler method traces out a solution by jumping forward in increments of Δx along arrows of the slope field diagram.

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

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

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:

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

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.

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 numerically integrating 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. It is nice to have a numerical updating scheme, but we have not investigated how closely the numerical solution approximates the actual solution. This is a big concern! Each step in the process has a error associated to it, so how do we guarantee after hundreds or thousands of iterations that the numerical solution is any good? 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 small 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).

3. 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.

4. (Not essential to know this for tests!) There is more than one way to derive the forward Euler formula. Consider the following three set-ups:

   (a) We are attempting to model the trajectory $y(x)$ satisfying $y'(x) = f(x, y)$. Consider the first-order Taylor series expansion of the trajectory at the point $x + \Delta x$ (assuming $0 < \Delta t \ll 1$). We have

$$\begin{aligned} y(x + \Delta x) &= y(x) + y'(x)\Delta x + O(\Delta x^2) \\ &= y(x) + f(x, y(x))\Delta x + O((\Delta x)^2). \end{aligned}$$

   Ignoring the terms of order $(\Delta x)^2$ and higher, this justifies the update scheme (1).

   (b) We are assuming that the derivative of the solution is given by $y'(x) = f(x, y(x))$ so, by definition, we have

$$\lim_{\Delta x} \frac{y(x + \Delta x) - y(x)}{\Delta x} = f(x, y(x)).$$

If we take Δx small enough, we have

$$\frac{y(x+\Delta x)-y(x)}{\Delta x} \approx f(x,y(x)) \quad \Longrightarrow \quad y(x+\Delta x) = y(x)+f(x,y(x))\Delta x.$$

This justifies (1).

(c) We could also notice that the equation $y'(x) = f(x,y(x))$ can be integrated to give

$$\int_0^t y'(x)\,dx = \int_0^t f(s,y(s))\,ds$$

$$\Longrightarrow \quad y(x) - y(0) = \int_0^x f(s,y(s))\,ds$$

$$\Longrightarrow \quad y(x) = y(0) + \int_0^x f(s,y(s))\,ds.$$

It remains to determine a numerical integration method for the integral on the right-hand side. We recall that integrals correspond to areas, so this just amounts to approximating the area under the curve $f(s,y(s))$ from $s=0$ to $s=x$. The easiest choice is the *rectangular rule*, which just approximates the area with a rectangle the width of the interval and the height given by one of the endpoints. In this case, we can choose

$$\int_0^x f(s,y(s))\,dx \approx f(x,y(x))\Delta x$$

which justifies the form (1).

It remains to consider how this formula actually performs. Consider the following example.

Example 1: Consider the initial value problem

$$\frac{dy}{dx} = y, \quad y(0) = 1.$$

We know that this system has the unique solution $y(x) = e^x$ but how close does the numerical solution come to it?

Let's consider the interval $x = 0$ to $x = 5$. Consider taking the step size $\Delta x = 1$ (that is to say, bumping the solution forward a full unit in each

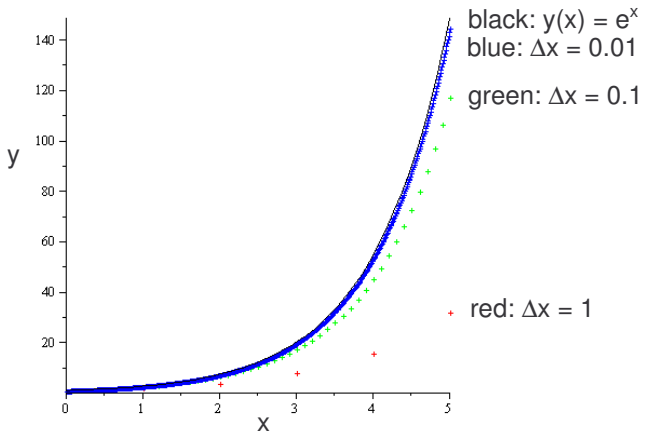

Figure 3: The forward Euler scheme with $\Delta x = 1$ (red), $\Delta x = 0.1$ (green) and $\Delta x = 0.01$ (blue). As the time-step decreases in size, the accuracy of the estimate to the true solution $y(x) = e^x$ increases, at the expense of needing greater computation resources (although all simulations ran in a fraction of a second on my laptop).

update). This necessitates taking $n = (x_{final} - x_{initial})/\Delta x = 5$ updates. This gives the sequence of poins:

$$y_{n+1} = y_n + f(x_n, y_n)\Delta x = y_n + y_n(1) = 2y_n$$

$$\begin{aligned} y_0 &= 1 \\ y_1 &= 2 \\ y_2 &= 4 \\ y_3 &= 8 \\ y_4 &= 16 \\ y_5 &= 32. \end{aligned}$$

That is to say, our numerical scheme has told us that $y(5) \approx 32$. We know, however, that $y(5) = e^5 \approx 148.4131591$. In other words, this approximation is not good at all. It is rather terrible, in fact.

We should not give up on the Euler method entirely, however. Consider taking $\Delta x = 0.1$ ($n = 50$) and $\Delta x = 0.01$ ($n = 500$). We can see that this significantly increases the accuracy of the numerical solution (see Figure 3)! In particular, we notice that, for $\Delta x = 0.1$ we have $y(5) \approx 117.3908529$, and

for $\Delta x = 0.01$ we have $y(5) \approx 144.7727724$.

Example 2: We have shown that decreasing the step Δx increases the accuracy of estimates. How else might we improve the accuracy of a numerical method?

The answer is that we can choose a different numerical method altogether. Consider the following method, which is a specified implementation of the *Runge-Kutta method*. (This method comes from approximating the integral in the third method for numerical approximation using the Simpson's Rule.) In this method, we define the quantities

$$\begin{aligned} k_1 &= f(x_n, y_n) \\ k_2 &= f(x_n + \frac{1}{2}\Delta x, y_n + \frac{1}{2}k_1\Delta x) \\ k_3 &= f(x_n + \frac{1}{2}\Delta x, y_n + \frac{1}{2}k_2\Delta x) \\ k_4 &= f(x_n + \Delta x, y_n + k_3\Delta x) \end{aligned}$$

and then update the system with

$$y_{n+1} = y_n + \frac{\Delta x}{6}\left(k_1 + 2k_2 + 2k_3 + k_4\right).$$

Using this method we can get a near exact approximate of the solution over the interval $x = 0$ to $x = 5$ using the time-step $\Delta x = 1$ (see Figure 4).

The moral of the story is that there is a trade-off between accuracy and computational resources in one direction or the other. Either we have to decrease the step-size, or we have to choose a more computationally-intensive numerical method. In general, it is some combination of both which is most effective.

It is also worth noting that the example we have considered was only illustrative, since we knew the exactly solution. In general practice no explicit solution is known by which to verify our numerical solution, so we must know that the method we using is sound. This is a very challenging (and exciting!) field of research but delving in any depth into it is beyond the scope of this course.

More examples are contained in Sections 2.4-2.6 of the text.

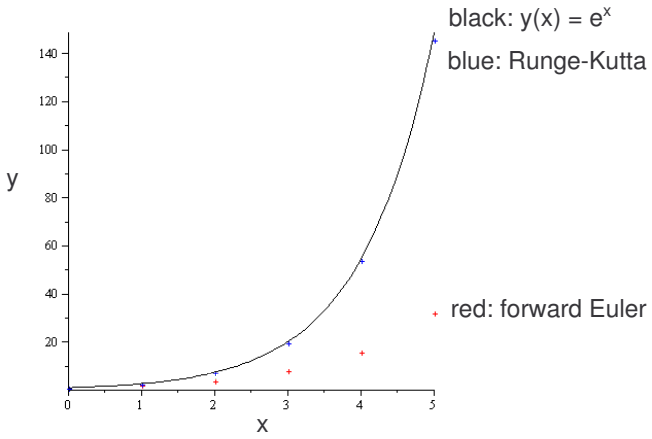

Figure 4: The forward Euler scheme with $\Delta x = 1$ (red) compared with the Runge-Kutta method (blue) with the same time step. We can see that agreement in the estimate to the true solution $y(x) = e^x$ increases by switching method, at the expense of needing greater computation resources in each step.