Remarks on ODEs

Sri Pranav Kunda

September 2022

1 Introduction

We discuss clarifications on two small topics in ordinary differential equations. We first start with Euler's method (RK1) and proceed to "correct" some things about the separation and change of variables.

To understand what RK1 is, it is good to have knowledge of a larger class of methods called the Runge Kutta Methods, which are techniques (algorithms) for finding solutions to ordinary differential equations.

Note that you've already learned about one of the Runge Kutta methods. Euler's method is the first-order Runge Kutta method, and it will be clear why it is the "first order method" once we discuss some of the other algorithms. Thus Euler's method is RK1, and the other methods are named similarly.

2 Euler's Method

Let's find an interesting (but terrible) way to compute π. Consider a function f such that $f(0) = 0$ and $f'(x) = 1/(1 + x^2)$ for any x.

Exercise: Find the function f which satisfies the above conditions.

The answer is obtained by integrating to find the specific solution to the differential equation, similar to what has already been demonstrated in class. Knowing that $f = \arctan$, computing $4f(1)$ should yield π.

The question now is, how do we compute $f(1)$ without actually knowing what f is?

What we do know from the definition of the derivative is that for small h such that $|h| > 0$ and any t_i,

$$f'(t_i) \approx \frac{f(t_i + h) - f(t_i)}{h} \iff hf'(t_i) + f(t_i) = f(t_i + h).$$

Let's call h the "step size" and let $t_0 = 0$, since we already know $f(0)$. Setting $t_k = t_0 + kh$, we can now approximate $f(t_k)$ for any positive integer k. In particular, for $w_0 = f(t_0) = 0$,

$$w_k = hf'(t_{k-1}) + w_{k-1} = \frac{h}{1+t_{k-1}^2} + w_{k-1}.$$

This is exactly what RK1, or Euler's method, does. The entire right hand side of the above expression is just a special case of the more general rule for w_k which is provided to the left.

Now, a more sensible use case might be when f' depends on f (and when it is non-separable). Well, we already know that w_k approximates $f(t_k)$. So, we may more generally claim that $g' = h(t, w)$, where t refers to some t_k and w refers to the approximation of $f(t_k)$. For example, if $g' = g$, then we say that $h(t_k, w_k) = w_k$ for all k.

Extra: Implementing RK1

For those who are not familiar with programming languages, this portion is not necessary. Using Python, we can quickly implement this algorithm. To demonstrate the recursive nature of the algorithm, I've implemented it once with recursion and another time with normal iteration methods. Recursion does not work as well in a practical sense due to limitations, especially concerning recursion depth for smaller h.

```
df = lambda x, f: 1/(1+x**2) # The derivative of arctan(x)

def euler_method_recursive(df, w_0, t_0, goal, h=10e-3):
    if t_0 < goal:
        return euler_method(df, h*df(t_0, w_0) + w_0, t_0 + h, goal)
    else:
        return w_0

def euler_method(df, w_k, t_k, goal, h=10e-8):
    while t_k <= goal:
      w_k += h*df(t_k, w_k)
      t_k += h
    return w_k

pi_approx = 4*euler_method(df, 0, 0, 1) # Our approximation of pi
```

For $h = 10^{-8}$, we receive `3.141592953718288` for the approximation of π. This approximation is correct for 6 decimal places. It quickly becomes clear that this is a really bad way to actually compute π. Click here for more efficient methods.

Exercise: Execute the computer program above to approximate e. You should use the `euler_method` function.

Extra: Constructing a Better Method

It is difficult to truly explain the "error analysis" part of Euler's method before first familiarizing yourself with some more mathematics. Namely, we want to look at derivatives of multivariate functions and the Taylor Series of a function. Thus this part is more mathematically advanced than the other parts of this section.

Definition 1 (Local Truncation Error)**.** *In the case of a general numerical algorithm, the local truncation error is the error induced by a single iteration of the algorithm.*

The error produced by approximating $y(t_k)$ from w_{k-1} (one iteration of Euler's method) is the local truncation error for Euler's method. We now proceed to compute this error when there is a "nice" solution to our ODE.

Suppose $y' = h(t, w)$ such that $y(t_0) = w_0$. Also assume the solution to this initial value problem y is unique, smooth, and has a convergent Taylor series in some ϵ-neighborhood of t_k (for a specific k) for $\epsilon > h$, where h is the step size. By Taylor's Theorem (See Rudin Pg. 110, Thm. 5.15 for proof), there exists a ξ between t_k and t_{k-1} so that

$$y(t_k) = \sum_{j=0}^{n-1} \frac{h^j y^{(j)}(t_{k-1})}{j!} + \frac{h^n y^{(n)}(\xi)}{n!}. \tag{1}$$

The $n = 2$ case yields

$$y(t_k) = y(t_{k-1}) + hy'(t_{k-1}) + \frac{h^2 y''(\xi)}{2}.$$

If $\epsilon > kh$, it follows that

$$y_k = w_k + O(h^2).$$

It is now clear mathematically that a smaller step size leads to a smaller local truncation error for the kth iteration, and that the error is proportional to h^2.

We can actually correct for this error. Note that

$$\frac{df(t,w)}{dt} = \left(\frac{\partial f}{\partial t} + \frac{\partial f}{\partial w} f\right)$$

so that

$$\frac{h^2 y''(t_{k-1})}{2} = \frac{h^2}{2}\left(\frac{\partial f}{\partial t} + \frac{\partial f}{\partial w} f\right)(t_{k-1}, y(t_{k-1}))$$

Then we may select $n = 3$ in 1 to receive

$$y(t_k) = y(t_{k-1}) + hy'(t_{k-1}) + \frac{h^2}{2}\left(\frac{\partial f}{\partial t} + \frac{\partial f}{\partial w} f\right)(t_{k-1}, y(t_{k-1}))$$

We may rewrite this as

$$y(t_k) = y(t_{k-1}) + \frac{h}{2} f(t_{k-1}, y(t_{k-1})) + \frac{h}{2}\left(f + h\frac{\partial f}{\partial t} + h\left(\frac{\partial f}{\partial w} f\right)\right)(t_{k-1}, y(t_{k-1})),$$

We end up with RK2 when we substitute the term in parentheses with the multivariate Taylor expansion, a second-order method (note that the error term is now smaller):

$$y(t_k) = y(t_{k-1}) + \frac{h}{2} f(t_{k-1}, y(t_{k-1})) + \frac{h}{2} f(t_k, y(t_{k-1}) + hf(t_{k-1}, y(t_{k-1}))) + O(h^3).$$

RK4 is constructed in a similar way.

3 Remark: Change/Separation of Variables

When we are taught to compute integrals with a u-substitution (a change of variables). However, in fact

$$\int f(u(x))u'(x)\,dx \neq \int f(u)\,du.$$

The key here is that the dx and du doesn't actually mean anything (especially in the case of computing antiderivatives). It is a notational convention, but you can replace u with x, t, or any other letter, and the expression represented by the integral will not change. However, we can definitely fix this. What we actually mean is

Theorem 1. *If F is an antiderivative of f, $\int f(g(x))g'(x)\,dx = F(g(x)) + C$.*

Proof. The proof follows from the chain rule and the definition of an antiderivative. □

To keep the convention going, we will abuse notation to assume $\int f(u)\,du := F(u(x)) + C$ so that $\int f(u(x))u'(x)\,dx = \int f(u)\,du$.

Another misconception is treating the separation of variables as "multiplying the infinitesimals" on both sides. That is, we treat $\frac{dy}{dx}$ as a ratio, where we can algebraically manipulate the numerator and denominator to find a solution to our differential equation.

You may recognize that this shouldn't actually work, since the derivative is the limit of a fraction, but not a fraction itself. dx or dy itself has no immediate meaning in this context.

However, there is a reason it works. This is called the separation of variables theorem. A friendly proof is provided below:

Theorem 2 (Separation of Variables)**.** *Suppose an ODE can be expressed as $y' = g(x)h(y)$. Then the following equality holds, where $\int f(t)\,dt = F$ means $\frac{dF}{dt} = f$:*

$$\int \frac{dy}{h(y)} = \int g(x)\,dx.$$

Proof. We may divide $h(y)$ and integrate on both sides to receive

$$\int \frac{dx}{h(y)}\frac{dy}{dx} = \int g(x)\,dx$$

With a change of variables (remember the abuse of notation we discussed above), we see that this is equivalent to

$$\int \frac{dy}{h(y)} = \int g(x)\,dx.$$

□