EULER AND PICARD'S METHOD

8

In this lecture, we present two methods for approximating the solution of an ODE. Euler's method is perhaps the simplest algorithm for numerical computations of solutions; other methods (some of which are pre-packaged in programs like Matlab) may be more sophisticated, but the basic ideas are often quite similar. Picard's Method is less of a numerical algorithm, and more as a theoretical tool. It will be used in the next lecture to prove the existence of solutions.

8.1 EULER'S METHOD

Consider the first-order nonlinear IVP

$$y' = f(x, y(x)) \quad \text{with} \quad y(x_0) = y_0, \tag{8.1}$$

with $x \in [x_0, x_N]$. If the value, $y(x)$ is known, then (8.1) is a prescription of the rate of change of $y(x)$, at any point, x. The idea of Euler's Method is to use this prescription to evolve the solution, starting from the initial point, $x = x_0$. In fact, you have already done so (graphically) when working with the direction fields.

When we work with numerical solutions, we need to divide the continuous interval $[x_0, x_N]$ into a discrete set of points. This process is often called *discretizing*, and the result is a *discretization* or *mesh*. Thus, we split the domain into N sub-intervals, with each sub-interval of the form, $[x_j, x_{j+1}]$, where $j = 0, 1, \ldots, N$. Our goal is to approximate the solution, $y(x)$, at the $N + 1$ points, using y_0, y_1, …, y_{N+1}.

Of course, the first point should simply be the initial value, y_0. In order to approximate the second point, y_1, we can apply Taylor's theorem to expand the function y about the point $x = x_0$:

$$y = y_0 + y'(x_0)(x - x_0) + \frac{y''(\xi)}{2}(x - x_0)^2, \tag{8.2}$$

Figure 8.1: Discretization

where $\xi \in [x_0, x]$. For x sufficiently close to x_0, we may use (8.2) as an approximation of the values of the solution. For the approximation at $x = x_1$, we thus have

$$y_1 \approx y_0 + f(x, y_0)(x_1 - x_0). \tag{8.3}$$

We'd like to repeat this procedure to construct the tangent line of the solution at $x = x_1$. The solution near $x = x_1$ satisfies

$$y = y(x_1) + f(x, y(x_1))(x_1 - x_0), \tag{8.4}$$

but unfortunately, we do not have the value of $y(x_1)$. The best we can do is to approximate this value using our previous approximation, $y(x_1) \approx y_1$. Thus for some point x_2 near x_1, we can use

$$y_2 \approx y_1 + f(x, y_1)(x_2 - x_1). \tag{8.5}$$

In general, then, we have for the j^{th} mesh point

$$y_j \approx y_{j-1} + f(x_j, y_{j-1})(x_j - x_{j-1}). \tag{8.6}$$

Suppose that we have chosen to discretize the domain $[x_0, x_N]$ into $N+1$ evenly spaced intervals of size h. It is clear from the Taylor series that the *local* error incurred by approximating $(x_1, y(x_1))$ using the first mesh point is,

$$\text{local error} \leq \frac{Mh^2}{2},$$

where M is the maximum of y'' over the interval $[x_0, x_1]$. What is the *global error* in proceeding step-by-step, using Euler's method, over the entire interval $[x_0, x_N]$? Since there are N total intervals, then an intuitive guess of the total global error might be:

$$\text{global error} \approx n\frac{\overline{M}h^2}{2} = (x_N - x_0)\frac{\overline{M}h}{2} = \mathcal{O}(h),$$

where $\overline{M}$ is the maximal value of y'' within the interval. There is a slight error in this reasoning, though, since continuing our approximation to $(x_2, y(x_2))$, note that there is also an additional error which is due to the fact that we have not used the value of the true slope, $y'(x_1)$, but instead have used $f(x_1, y_1)$ which is already off by an amount of order h. However, it can be rigorously shown that the above estimate is 'close', and that the global error is indeed of order h. The consequence of this is that if we double the number of points used in Euler's method, the error should halve.

8.2 Improved Euler's method

Can we do better than Euler's method? One way is to continue to higher orders in Taylor's approximation, but this leads to the problem of determining higher derivatives. One alternative viewpoint is given by reposing the IVP

$$y' = f(x, y), \qquad y(x_0) = y_0,$$

as an integral equation (a technique we shall come back to often). If we integrate both sides of the equation from x_0 to x_1, this gives

$$y(x_1) = y(x_0) + \int_{x_0}^{x_1} f(s, y(s))\, ds.$$

Notice that the unknown function $y(x)$ appears on both sides of the equation. If we approximate the integral using the Riemann sum (a box), then we have

$$y(x_1) \approx y_1 = y_0 + f(x_0, y_0)(x_1 - x_0),$$

which is simply Euler's method. But what happens if we approximate the integral using a trapezoid, with

$$y(x_1) \approx y_1 = y_0 + \left(\frac{f(x_0, y_0) + f(x_1, y_1)}{2} \right) h.$$

As we showed earlier in class, this method yields an $\mathcal{O}(h^2)$ error instead of simply an $\mathcal{O}(h)$ error. The difficulty, however, is that y_1 appears on both sides of the equation. The idea behind the *Improved Euler method* is to use Euler's approximation for y_1 on the right hand-side, and then compute a new value of y_1 according to the above formula. Thus we have the following algorithm:

Algorithm 8.1 (Improved Euler Method). *Given an initial condition $y(x_0) = y_0$ and step size, h, compute the point (x_j, y_{j+1}) using the point (x_j, y_j) in the following manner:*

1. *Compute the slope $m_j = f(x_j, y_j)$*
2. *Use Euler's method to compute $\tilde{y}_{j+1} = y_j + f(x_j, y_j)h$*
3. *Compute the slope $n_j = f(x_{j+1}, \tilde{y}_{j+1})$*
4. *The final quantity is approximated by*

$$y_{j+1} = y_j + \left(\frac{m_j + n_j}{2} \right) h.$$

The Improved Euler's method is called a **predictor-corrector** method, because it predicts the value of y_{j+1} using $\tilde{y}_{j+1}$, and then uses this value to correct the solution. It can be shown that the global error in using the Improved Euler method is $\mathcal{O}(h^2)$, so it is one order better than the regular Euler's method.