Ordinary Differential Equations (Ode's)

We shall consider some numerical schemes to solve initial-value problems (Cauchy's problems) written as: Find $y = y(t)$ solution of

$$\begin{cases} y'(t) = f(t, y(t)) & t \in [t_0, T] \\ y(t_0) = y_0. \end{cases} \tag{1}$$

We assume $y : [t_0, T] \to \mathbb{R}$ but can be generalized to $y : [t_0, T] \to \mathbb{R}^d$

In general $f(t, y(t))$ is a non-linear function describing the evolution in time of $y(t)$. The true solution $y(t)$ of (1) evolves continuously in time, and we want to follow it by a discrete approximation.

Both exact and discrete solution of (1) start from the same initial value y_0 at t_0. The discrete one takes finite steps Δt, and after n steps it reaches a value y_n. We hope and expect that y_n is close to the exact value $y(t_0 + n\Delta t)$. We shall see that this may or may not happen.

Numerical methods for Ode's

Let us see some schemes to solve numerically (1). They are numerous, and a first distinction is among 1-step methods and multi-step methods. Let us see 1-step methods. They can all be derived in the following way.

Le $t_0, t_1, \cdots, t_N = T$ be a set of points in $[t_0, T]$; as usual, for simplify notation we take them equally spaced: (N given, we define $\Delta t = \dfrac{T - t_0}{N}$ and we set $t_0, t_1 = t_0 + \Delta t, t_2 = t_1 + \Delta t, \cdots, t_N = T$).
At each step, (on each subinterval $[t_n, t_{n+1}]$) *we integrate the differential equations...*

$$y'(t) = f(t, y(t))$$

$$y(t_{n+1}) - y(t_n) = \int_{t_n}^{t_{n+1}} y'(t)dt = \int_{t_n}^{t_{n+1}} f(t, y(t))dt. \qquad (*)$$

Then, as $y(t)$ in the interval $[t_n, t_{n+1}]$ is unknown to us (and moreover, in general, we are unable to compute the integral exactly), we use some quadrature formula.

$$y(t_{n+1}) - y(t_n) = \int_{t_n}^{t_{n+1}} y'(t)dt = \int_{t_n}^{t_{n+1}} f(t, y(t))dt. \qquad (*)$$

$$\int_{t_n}^{t_{n+1}} f(t, y(t))dt \sim \text{ quadrature formula.}$$

Different choices of quadrature formulas give rise to different schemes.

Examples of numerical schemes

Example 1 We consider first the quadrature formula

$$\int_c^d g(s)ds \simeq (d-c)\,g(c) \tag{2}$$

that, indeed, is very poor (and is exact only for $g = constant$). Then

$$\int_{t_n}^{t_{n+1}} f(t,y(t))dt \simeq (t_{n+1}-t_n)f(t_n,y(t_n)) \quad n = 0,1,2,\cdots \tag{3}$$

Using (3) into $y(t_{n+1}) - y(t_n) = \int_{t_n}^{t_{n+1}} y'(t)dt = \int_{t_n}^{t_{n+1}} f(t,y(t))dt$ we get:

$$y(t_1) \simeq y(t_0) + \Delta t\, f(t_0, y(t_0)) = y_0 + \Delta t\, f(t_0,y_0) =: y_1$$
$$y(t_2) \simeq y(t_1) + \Delta t\, f(t_1, y(t_1)) \simeq y_1 + \Delta t\, f(t_1,y_1) =: y_2$$
$$\vdots$$
$$y(t_N) \simeq y(t_{N-1}) + \Delta t\, f(t_{N-1}, y(t_{N-1})) \simeq y_{N-1} + \Delta t\, f(t_{N-1},y_{N-1}) =: y_N$$

Examples of numerical schemes

It is clear from this that errors accumulate at each step and might produce unexpected results. We will analyse the scheme later on. Let us write it in a compact form:

$$\begin{cases} y_0 \text{ given} \\ y_{n+1} = y_n + \Delta t\, f(t_n, y_n) \quad n = 0, 1, \cdots, N-1 \end{cases} \qquad (EE)$$

This is called EXPLICIT EULER method or FORWARD EULER method: at each step, the value y_n can be explicitly computed using values at the previous steps. It is very simple and inexpensive but, as we shall see, there is a "but"...

Examples of numerical schemes

Example 2 This time we consider the quadrature formula

$$\int_c^d g(s)ds \simeq (d-c)\,g(d) \tag{4}$$

that is also very poor and is exact only if $g = constant$, like the previous one. However the resulting scheme will be very different. In fact, applying it to our case we get

$$\int_{t_n}^{t_{n+1}} f(t,y(t))dt \simeq (t_{n+1}-t_n)f(t_{n+1},y(t_{n+1})) \quad n = 0,1,2,\cdots$$

that used into $y(t_{n+1}) - y(t_n) = \int_{t_n}^{t_{n+1}} y'(t)dt = \int_{t_n}^{t_{n+1}} f(t,y(t))dt$ gives

$$\begin{aligned}
y(t_1) &\simeq y(t_0) + \Delta t\, f(t_1, y(t_1)) \simeq y_0 + \Delta t\, f(t_1, y_1) =: y_1 \\
y(t_2) &\simeq y(t_1) + \Delta t\, f(t_2, y(t_2)) \simeq y_1 + \Delta t\, f(t_2, y_2) =: y_2 \\
&\vdots \\
y(t_N) &\simeq y(t_{N-1}) + \Delta t\, f(t_N, y(t_N)) \simeq y_{N-1} + \Delta t\, f(t_N, y_N) =: y_N
\end{aligned}$$

Examples of numerical schemes

The scheme becomes

$$\begin{cases} y_0 \text{ given} \\ y_{n+1} = y_n + \Delta t\, f(t_{n+1}, y_{n+1}) \quad n = 0, 1, \cdots, N-1 \end{cases} \qquad (IE)$$

This is called IMPLICIT EULER method or BACKWARD EULER method. Note that, at every time step, the unknown y_{n+1} in (IE) appears both on the left-hand side *and in the right-hand side*, and in order to perform the step we must *solve an equation* in the unknown y_{n+1}. Since f is in general non-linear, at each step, to find y_n we need to solve a non-linear equation (for example, with Newton method). The method is obviously more expensive than Explicit Euler.

Examples of numerical schemes

Example 3 As a third example we consider the quadrature formula

$$\int_c^d g(s)ds \simeq (d-c)\left(\frac{g(c)+g(d)}{2}\right) \tag{5}$$

(trapezoidal rule) that is better than the previous ones since it is exact whenever g is a polynomial of degree ≤ 1. Applying it to our case we get

$$\int_{t_n}^{t_{n+1}} f(t,y(t))dt \simeq \frac{(t_{n+1}-t_n)}{2}\Big(f(t_n,y(t_n))+f(t_{n+1},y(t_{n+1}))\Big) \quad \forall n$$

The corresponding scheme becomes

$$\begin{cases} y_0 \text{ given} \\ y_{n+1} = y_n + \dfrac{\Delta t}{2}\Big(f(t_n,y_n)+f(t_{n+1},y_{n+1})\Big) \; n = 0,1,\cdots,N-1 \end{cases}$$

Examples of numerical schemes

$$\begin{cases} y_0 \text{ given} \\ y_{n+1} = y_n + \dfrac{\Delta t}{2}\Big(f(t_n, y_n) + f(t_{n+1}, y_{n+1})\Big) \; n = 0, 1, \cdots, N-1 \end{cases}$$

This is called CRANK-NICOLSON method. It is an implicit method (and hence, as the previous Implicit Euler, expensive) but it has a good accuracy, as we shall see.

Examples of numerical schemes

Example 4 If we replace, in the Crank-Nicolson scheme, y_{n+1} with $y^*_{n+1} = y_n + \Delta t\, f(t_n, y_n)$, that is, with the value predicted by Explicit Euler, we get rid of the implicit part and obtain a new explicit method, called HEUN method, which reads

$$\begin{cases} y_0 \text{ given} \\ y^*_{n+1} = y_n + \Delta t\, f(t_n, y_n) \qquad (HEUN) \\ y_{n+1} = y_n + \dfrac{\Delta t}{2}\Big(f(t_n, y_n) + f(t_{n+1}, y^*_{n+1})\Big) \quad n = 0, 1, \cdots, N-1 \end{cases}$$

So we have two classes: explicit methods, and implicit methods.
In all cases we want the sequence $\{y_0, y_1, \cdots, y_N\}$ to converge to the sequence $\{y_0, y(t_1), \cdots, y(T)\}$.

If, given a method, we can prove that

$$\exists C > 0 \text{ such that } \quad \max_n |y_n - y(t_n)| \leq C\Delta t^p$$

with C independent of Δt and $p > 0$, then we say that *the method is convergent*, and *the order of convergence is p* (the bigger p, the faster the convergence).