ODE SOLVERS: ONE-STEP METHODS

LONG CHEN

ABSTRACT. This is my brief summary of Chapter 5 and 6 of the book *Numerical Analysis* by *Gautschi, Walter*. Discussion is simplified and more figures are provided for a better illustration.

CONTENTS

We consider numerical methods for solving the nonlinear ODE

$$y' = f(t, y), \quad y(a) = y_0, \tag{1}$$

where $t \in \mathbb{R}$ is the independent variable, $y = y(t) \in \mathbb{R}^d$ may be a vector-valued function, and the function $f(t, y)$ is given. Assume $f(t, y)$ is Lipschitz continuous with respect to y, i.e., $\|f(t, y_1) - f(t, y_2)\| \leq L\|y_1 - y_2\|$. Under this condition, the solution y to (1) exists and is unique, at least in a neighborhood of a. We further assume that the solution exists for all $t \in \mathbb{R}$. The focus here is on how to compute a numerical approximation of y.

A noticeable difference from the textbook *Numerical Analysis* by *Gautschi, Walter* is that the independent variable is changed from x to t, which more naturally represents time.

1. SCHEMES

1.1. **Forward Euler Method.** Besides the initial value y_0, equation (1) also provides the initial derivative $y'(a) = f(a, y_0)$. Using the Taylor expansion, we obtain

$$y(a + h) \approx y(a) + hy'(a) = y_0 + hf(a, y_0).$$

Date: January 14, 2025.

To clearly distinguish between the exact solution y to (1) and its numerical approximation, we introduce the notation u_n for the numerical approximation, where the subscript n corresponds to the grid point t_n. The step size, which may vary at each iteration, is denoted by h_n.

The simplest forward (explicit) Euler method can be summarized as follows:

Forward/Explicit Euler Method

Let $t_0 = a$, $u_0 = y_0$. For $n = 0, 1, 2, \ldots,$

$$t_{n+1} = t_n + h_n,$$
$$u_{n+1} = u_n + h_n f(t_n, u_n).$$

Geometrically, the exact solution y to (1) is a curve, and the Euler method approximates this curve using a polygon; see Fig. 1. Intuitively, as the length of each side of the polygon approaches zero, the numerical approximation can closely approximate the true solution. Numerical analysis provides a mathematical framework to rigorously analyze the error and, when possible, improve the methods.

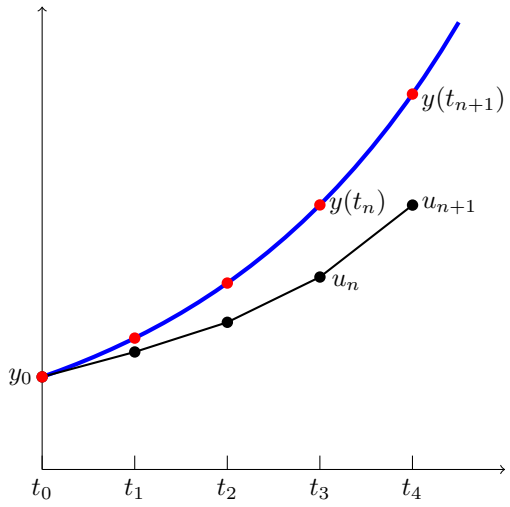

FIGURE 1. An example of one step method.

Question: Euler method can be understood as moving along the tangential direction with a small step size. Is there any better direction to move?

1.2. **One step method and truncation error.** We first provide a measure of how "good" a method is. Consider a single interval $[t_n, t_{n+1}]$. The step size $h_n := t_{n+1} - t_n$ may vary but is usually uniform. A generic form of a one-step method is given by

$$u_{n+1} = u_n + h_n \Phi(t_n, u_n; h_n), \tag{2}$$

where $\Phi(t_n, u_n; h_n)$ represents the direction of movement starting from (t_n, u_n). For the forward Euler method, $\Phi(t_n, u_n; h_n) = f(t_n, u_n)$.

Now consider the ODE (1), $y' = f(t, y)$, restricted to the interval $[t_n, t_{n+1}]$ with the initial condition $y(t_n) = y_n$. The exact solution value at t_{n+1} is denoted by $y_{n+1} = y(t_{n+1})$.

Assume $u_n = y_n$ and use the error $u_{n+1} - y_{n+1}$ as a measure of accuracy. It is easy to see that this error is proportional to the step size. To provide a more consistent comparison, we define the normalized quantity

$$T(t_n, y_n; h_n) := \frac{1}{h_n}(u_{n+1} - y_{n+1}),$$

which is referred to as the *truncation error*.

For simplicity, we omit the subscript n and use the subscript $_{\text{next}}$ to indicate $n+1$. With this notation, the truncation error can be rewritten as

$$T(t, y; h) = \frac{1}{h}(u_{\text{next}} - y_{\text{next}}).$$

This relationship is summarized in the following figure.

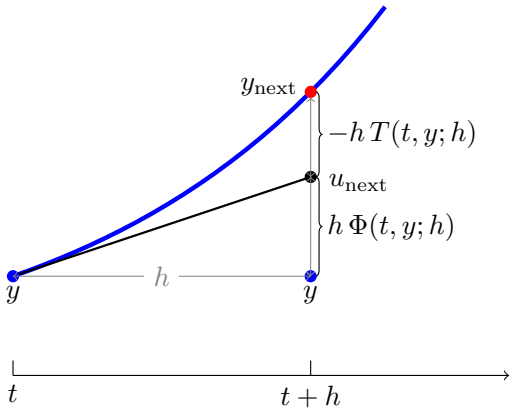

FIGURE 2. One step method and its truncation error.

Note that we add a negative sign in front of the truncation error in Fig. 2. The reason is that the length of a line segment is always positive, but by the definition of T, it can take negative values. For the example in the figure, T is negative, and $-hT$ represents the distance.

Definition 1.1 (Order of a method)**.** *The method is said to have order p if, for some vector norm $\|\cdot\|$,*

$$\|T(t, y; h)\| \le Ch^p, \tag{3}$$

uniformly on $[a, b] \subset \mathbb{R}^d$, with a constant C not depending on t, y, or h. We express this property briefly as

$$T(t, y; h) = \mathcal{O}(h^p), \quad h \to 0. \tag{4}$$

Note that $p > 0$ implies consistency. Usually, p is an integer ≥ 1. It is called the exact order *if (3) does not hold for any larger p.*

Definition 1.2 (Principal error function)**.** *A function* $\tau : [a,b] \subset \mathbb{R}^d \to \mathbb{R}^d$ *that satisfies* $\Psi(t,y) \neq 0$ *and*

$$T(t,y;h) = \tau(t,y)h^p + \mathcal{O}(h^{p+1}), \quad h \to 0, \tag{5}$$

is called the principal error function*.*

Using the definition of the one-step method (cf. (2)) and the fact that $y' = f(t,y)$, the truncation error can also be expressed as

$$T(t,y;h) = \Phi(t,y;h) - \frac{1}{h}\int_t^{t+h} f(s,y(s))\,\mathrm{d}s. \tag{6}$$

This shows that the truncation error represents the discrepancy between the numerical quadrature $\Phi(t,y;h)$ and the average of $f(t,y(t))$ over the interval $[t,t+h]$.

From this perspective, we can immediately see that the forward Euler method has a first-order truncation error:

$$T_{\mathrm{Euler}}(t,y;h) = f(t,y(t)) - \frac{1}{h}\int_t^{t+h} f(s,y(s))\,\mathrm{d}s = \mathcal{O}(h).$$

If we use the midpoint rule or the trapezoidal rule for the integral, the order of accuracy improves to second order. In general, numerical quadrature rules of arbitrary order can be applied.

For instance, by performing a Taylor expansion at $t + h/2$, we find:

$$f\left(t+\frac{h}{2}, y\left(t+\frac{h}{2}\right)\right) - \frac{1}{h}\int_t^{t+h} f(s,y(s))\,\mathrm{d}s = \mathcal{O}(h^2).$$

What is the extra difficulty here compared to standard numerical quadrature?

1.3. **Second-order methods.** The difference and difficulty lie in the fact that $y(t+\frac{h}{2})$ is unknown; we only have the initial value $y(t)$. How can we address this? One approach is to use the forward Euler method to advance by $h/2$ and obtain u_{half} as an approximation of $y(t+\frac{h}{2})$. Specifically, we approximate $f(t+\frac{h}{2}, y(t+\frac{h}{2}))$ with $f(t+\frac{h}{2}, u_{\mathrm{half}})$. This leads to the update formula:

$$u_{\mathrm{next}} = u + hf\left(t+\frac{h}{2}, u + \frac{h}{2}f(t,u)\right).$$

In practice, we can unfold the nested expressions of f to derive the following improved Euler method:

Improved Euler Method

Let $t_0 = a$, $u_0 = y_0$. For $n = 0, 1, 2, \ldots,$

$$\begin{aligned} k_1 &= f(t_n, u_n), \\ k_2 &= f(t_n + h_n/2, u_n + k_1 h_n/2), \\ u_{n+1} &= u_n + h_n k_2. \end{aligned}$$

The direction k_2 is an improvement over k_1 because it reduces the truncation error to $\mathcal{O}(h^2)$.

Similarly, by modifying the trapezoidal method for numerical quadrature, we obtain a method known as the Heun method. It is also a second-order method.

Heun Method

Let $t_0 = a$, $u_0 = y_0$. For $n = 0, 1, 2, ...,$

$$k_1 = f(t_n, u_n),$$
$$k_2 = f(x_n + h_n, u_n + k_1 h_n),$$
$$u_{n+1} = u_n + h_n(k_1 + k_2)/2.$$

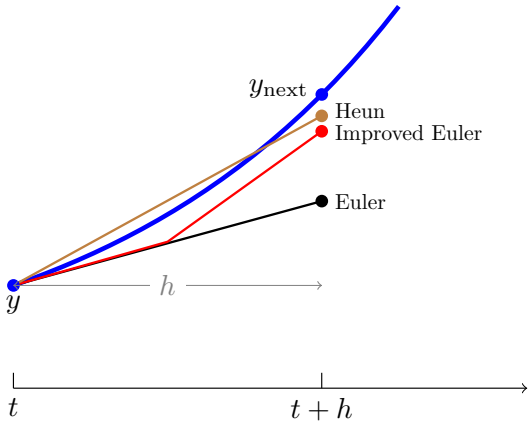

FIGURE 3. Forward Euler, improved Euler and Heun methods

Notice that both the modified Euler method and Heun's method require evaluating the slope twice; see Fig. 3. These are special cases of two-stage methods, which will be discussed below.

1.4. **Two-stage methods.** A generic form of two-stage methods is given by

$$u_{\text{next}} = u + (\alpha_1 h)k_1 + (\alpha_2 h)k_2, \tag{7}$$

where the weight $\alpha_1 \in [0, 1)$, and $\alpha_1 + \alpha_2 = 1$. The geometric interpretation is that we first move along the direction k_1 with a step size of $\alpha_1 h$ and then switch to a better direction k_2 for the remaining step size.

We introduce another parameter $\mu \in (0, 1]$ and treat k_2 as an approximation of the slope at $(t + \mu h, y(t + \mu h))$. Since $y(t + \mu h)$ is unknown, it is approximated using the forward Euler method, i.e., $k_2 = f(t + \mu h, y + \mu h k_1)$.

Next, we aim to choose the parameters $(\alpha_1, \alpha_2, \mu)$ to minimize the truncation error as formulated in (11). The primary tool for this is a Taylor expansion at the left endpoint (t, y). First,

$$\begin{aligned} k_2 &= f(t + \mu h, y + \mu h k_1) \\ &= f + \mu h f_t + \mu h k_1 f_y + \frac{1}{2}\left[\mu^2 h^2 f_{tt} + 2\mu^2 h^2 f_{ty} k_1 + \mu^2 h^2 k_1^{\intercal} f_{yy} k_1\right] + \mathcal{O}(h^3). \end{aligned}$$

Recall that $k_1 = f = f(t,y)$. Substituting this into the expression for $\Phi(t,y;h)$, we obtain

$$\begin{aligned}\Phi(t,y;h) &= \alpha_1 k_1 + \alpha_2 k_2 \\ &= f + \alpha_2 \mu h (f_t + f_y f) + \frac{\alpha_2}{2}\mu^2 h^2 \left[f_{tt} + 2 f_{ty} f + f^\intercal f_{yy} f\right] + \mathcal{O}(h^3).\end{aligned}$$

Then we compare $\Phi(t,y;h)$ with $\frac{1}{h}\int_t^{t+h} f(s,y(s))\,\mathrm{d}s$. We expand the integrand $f(s,y(s))$ at (t,y):

$$f(s,y(s)) = f + \frac{\mathrm{d}f}{\mathrm{d}s}(s-t) + \frac{1}{2}\frac{\mathrm{d}^2 f}{\mathrm{d}s^2}(s-t)^2 + \mathcal{O}(h^3). \tag{8}$$

Since the second variable y is also a function of the first one, the derivatives, by the chain rule, are given as

$$\begin{aligned} f^{[1]}(t,y(t)) &:= \frac{\mathrm{d}f}{\mathrm{d}t}(t,y(t)) = f_t + f_y y' = f_t + f_y f, \\ f^{[2]}(t,y(t)) &:= \frac{\mathrm{d}f^{[1]}}{\mathrm{d}t}(t,y(t)) = f_t^{[1]} + f_y^{[1]} f \\ &= f_{tt} + 2 f_{ty} f + f^\intercal f_{yy} f + f_y f^{[1]}. \end{aligned}$$

Higher derivatives $f^{[k]} := \frac{\mathrm{d}^k f}{\mathrm{d}t^k}$ can be computed recursively.

Substituting (8) into $h^{-1}\int_t^{t+h} f(s,y(s))\,\mathrm{d}s$, and combining this with the expansion of Φ, we obtain

$$\begin{aligned} T(t,y;h) =& (\alpha_1 + \alpha_2 - 1) f + \left(\alpha_2\mu - \frac{1}{2}\right) h f^{[1]} - \frac{1}{6} h^2 f_y f^{[1]} \\ &+ \frac{1}{2} h^2 \left(\alpha_2 \mu^2 - \frac{1}{3}\right)\left(f_{tt} + 2 f_{ty} f + f^\intercal f_{yy} f\right) + \mathcal{O}(h^3). \end{aligned}$$

Since the red term is generally nonzero, the two-stage methods are limited to at most second-order accuracy. We achieve second-order accuracy by choosing parameters that satisfy the relations

$$\alpha_1 + \alpha_2 = 1, \quad \alpha_2 \mu = \frac{1}{2}.$$

Examples of $(1-\alpha_2, \alpha_2, 1/(2\alpha_2))$ include

- Improved Euler method: $(0, 1, 1/2)$;
- Heun method: $(1/2, 1/2, 1)$;
- The choice $(1/4, 3/4, 2/3)$ will remove one part in h^2 term.

Notice that the choice $(1,0,0)$ corresponds to the forward Euler method, which only achieves a truncation order of $\mathcal{O}(h)$.

1.5. **Runge-Kutta methods.** A generalization of two-stage methods is r-stage Runge-Kutta methods:

$$\begin{aligned} \Phi(t,y;h) &= \sum_{s=1}^{r} \alpha_s k_s, \\ k_1 &= f(t,y); \\ k_s &= f\Big(t + \mu_s h, y + h\sum_{j=1}^{s-1} \lambda_{sj} k_j\Big), \quad s = 2,3,\dots,r. \end{aligned}$$

The maximal order of r-stage R-K method is bounded by r and only achievable for $r \le 4$. Systematical way of finding parameters can be found in ... Here we only list a popular one.

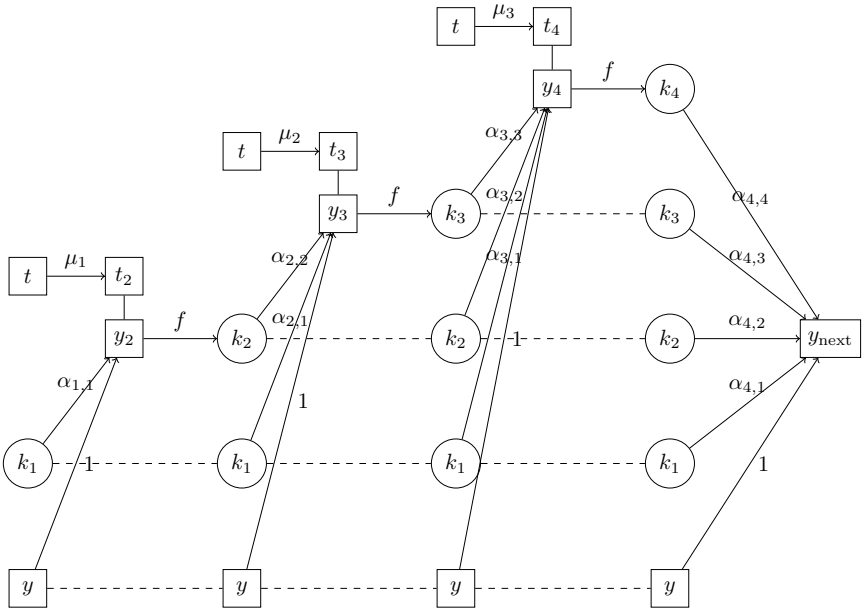

FIGURE 4. Runge-Kutta methods

The classical 4-th order Runge-Kutta method

Let $t_0 = a$, $u_0 = y_0$. For $n = 0, 1, 2, ...,$

$$\begin{aligned} k_1 &= f(t_n, u_n), \\ k_2 &= f(t_n + \frac{1}{2}h_n, u_n + \frac{1}{2}h_n k_1), \\ k_3 &= f(t_n + \frac{1}{2}h_n, u_n + \frac{1}{2}h_n k_2), \\ k_4 &= f(t_n + h_n, u_n + h_n k_3), \\ u_{n+1} &= u_n + \frac{h_n}{6}(k_1 + 2k_2 + 2k_3 + k_4). \end{aligned}$$

It resembles Simpson's rule for numerical quadrature:

$$\int_t^{t+h} f(s)\,\mathrm{d}s \approx \frac{h}{6}\left(f(t) + 4f(t + h/2) + f(t + h)\right).$$

When the integrand is changed to $f(s, y(s))$, approximations of $y(t + h/2)$ and $y(t + h)$ are required. Verifying the order involves a tedious Taylor expansion.

2. CONVERGENCE ANALYSIS

We recall the generic one step method below.

One Step Method (Φ)

Let $t_0 = a$, $u_0 = y_0$. For $n = 0, 1, 2, ...,$

$$t_{n+1} = t_n + h_n,$$
$$u_{n+1} = u_n + h_n \Phi(t_n, u_n; h_n).$$

2.1. **Error equation.** We first set up the space with norm and the operators. We partition the interval $[a, b]$ into a uniform grid $\mathcal{T}_h$ with size h

$$a = t_0 < t_1 < \cdots < t_N = b, \quad t_i = a + ih, h = (b - a)/N.$$

Introduce the vector space $\Gamma_h = \{\boldsymbol{v} = (v_0, v_1, \ldots, v_N)\} \cong \mathbb{R}^{N+1}$ equipped with $\|\cdot\|_\infty$ norm. As h is fixed, we will write $\Phi(t_n, u_n; h_n)$ as $\Phi_h(t_n, u_n)$ to simplify the notation and emphasize the variables are (t, y).

For $v \in C^1[a, b]$, define the residual operator $Rv = v'(t) - f(t, v(t))$. Similarly, define the discrete residual operator

$$(R_h \boldsymbol{v})_n = \frac{1}{h}(v_{n+1} - v_n) - \Phi_h(t_n, v_n), \quad n = 0, 1, \ldots, N - 1.$$

Then by definition, we have the equations

$$\begin{aligned} Ry &= 0, & y(0) &= y_0 \\ R_n \boldsymbol{u} &= 0, & u_0 &= y_0. \end{aligned}$$

where y is the solution to the initial value problem.

To measure the error, we need to put y and $\boldsymbol{u}$ into the same space. For a function $v \in C[a, b]$, introduce the nodal interpolation $\boldsymbol{v}_I \in \Gamma_h$ as

$$v_n = v(t_n), \quad n = 0, 1, \ldots, N.$$

Notice that v as a continuous function is defined everywhere in the interval while $\boldsymbol{v}_I$ is a vector with function values defined only on grid points.

Now we are ready to present the error equation.

$$R_h(\boldsymbol{u} - \boldsymbol{y}_I) = R_h \boldsymbol{y}_I = -T(\boldsymbol{y}_I; h). \tag{9}$$

2.2. **Stability.** We introduce the stability of the operator R_h. Consider the equation

$$\begin{aligned} R_h \boldsymbol{w} &= \boldsymbol{\varepsilon} \\ w_0 &= \eta_0. \end{aligned}$$

If there exists a constant $K > 0$ independent of h s.t.

$$\|\boldsymbol{w}\|_\infty \leqslant K\left(\|\eta_0\| + \|\boldsymbol{\varepsilon}\|_\infty\right),$$

we call R_h is stable.

Theorem 2.1. *If $\Phi(t, y; h)$ satisfies a Lipschitz condition with respect to the y-variables,*

$$\|\Phi_h(t, y) - \Phi_h(t, y^*)\| \leq M\|y - y^*\| \text{ on } [a, b] \times \mathbb{R}^d \times [0, h_0]$$

then the one step method defined by Φ is stable.

We precede the proof with the following useful lemma. The proof is elementary.

Lemma 2.2. *Lemma 5.7.1. Let $\{e_n\}$ be a sequence of numbers $e_n \in \mathbb{R}$ satisfying*

$$e_{n+1} \le a_n e_n + b_n, \quad n = 0, 1, \ldots, N-1$$

where $a_n > 0$ and $b_n \in \mathbb{R}$. Then

$$e_{n+1} \le E_{n+1}, \text{ with}$$
$$E_{n+1} = \left(\prod_{k=0}^{n} a_k\right) e_0 + \sum_{k=0}^{n} \left(\prod_{\ell=k+1}^{n} a_\ell\right) b_k, n = 0, 1, \ldots, N$$

Here is a graphical representation of the coefficients

$$\begin{pmatrix} a_0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ a_1 & a_1 & 0 & \cdots & 0 & 0 & 0 \\ a_2 & a_2 & a_2 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ a_n & a_n & a_n & \cdots & a_n & a_n & 1 \\ \\ e_0 & b_0 & b_1 & b_2 & \cdots & b_{n-1} & b_n \end{pmatrix}$$

We need to control the product $\prod_{k=0}^{n} a_k \le C$ uniformly to n. Fortunately $a_k = 1 + Mh$ is a perturbation near 1. Also there is accumulation of b_n sequence $\sum_{k=0}^{N} |b_k|$.

2.3. **Convergence.** We write out the error equation (9) in detail:

$$\begin{aligned} &u_{n+1} - y(t_{n+1}) - (u_n - y(t_n)) \\ &= h\left[\Phi_h(t_n, u_n) - \Phi_h(t_n, y(t_n))\right] - hT_h(t_n, y(t_n)). \end{aligned} \tag{10}$$

Let $e_{n+1} = |u_{n+1} - y(t_{n+1})|$ and $b_n = h|T_h(t_n, y(t_n))|$. Using the Lipschitz continuity, we then get the inequality

$$e_{n+1} \le (1 + Mh)e_n + b_n.$$

Then $a_k = 1 + Mh$ and

$$\prod_{k=0}^{N-1} a_k = (1 + Mh)^N = \left(1 + \frac{M(b-a)}{N}\right)^N \le e^{(b-a)M}.$$

Assuming $\|T\|_\infty \le Ch^p$, the accumulation of b_k is controlled as

$$\sum_{k=0}^{N} \left(\prod_{\ell=k+1}^{N} a_\ell\right) b_k \le Ch^p e^{(b-a)M} \sum_{k=0}^{N} h = Ch^p e^{(b-a)M}(b-a).$$

The extra factor h is used to ensure the accumulation is bounded by the length of interval.

We summarize the convergence in the following theorem.

Theorem 2.3. *If $\Phi_h(t, y)$ is Lipschitz with respect to the y-variables, and the truncation error $\|T_h(t_n, y(t_n))\|_\infty = \mathcal{O}(h^p)$, then*

$$\|\boldsymbol{u} - \boldsymbol{y}_I\|_\infty = \mathcal{O}\left(h^p\right) \text{ as } |h| \to 0.$$

3. Error Expansion

We assume the truncation error

$$T_h(t,y) := \frac{1}{h}(u_{\text{next}} - y_{\text{next}}) = \Phi_h(t,y) - \frac{1}{h}\int_t^{t+h} f(s,y(s))\,\mathrm{d}s \tag{11}$$

have the expansion

$$T_h(t,y) = \tau(t,y)h^p + \mathcal{O}(h^{p+1}),$$

where $\tau(t,y)$ is called the principle error function (of the truncation error).

We want the principle error function for the total error

$$u_n - y(t_n) = e(t_n)h^p + \mathcal{O}(h^{p+1}).$$

What is the difference? When definiting the truncation error, u_{next} and y_{next} share the same initial condition at t_n while u_{n+1} and $y(t_{n+1})$ are not. The truncation error is accumlated.

We claim the principal error function satisfies the following ODE.

Theorem 3.1. *Let $e(x)$ be the solution of the linear initial value problem*

$$\begin{aligned} e'(t) &= f_y(t,y(t))e + \tau(t,y(t)), \quad a \le t \le b,\\ e(a) &= 0. \end{aligned}$$

Then, for $n = 0, 1, \ldots, N$,

$$u_n - y\left(t_n\right) = e\left(t_n\right)h^p + O\left(h^{p+1}\right) \text{ as } h \to 0.$$

Proof. We use Taylor series to expand

$$\begin{aligned} \Phi_h(t_n,u_n) - \Phi_h(t_n,y(t_n)) &= \partial_y\Phi_h(t_n,y(t_n))(u_n - y(t_n))\\ &= \partial_y\Phi_0(t_n,y(t_n))(u_n - y(t_n)) + \mathcal{O}(h^{p+1})\\ &= f_y(t_n,y(t_n))(u_n - y(t_n)) + \mathcal{O}(h^{p+1}). \end{aligned}$$

To distinguish the continuous error function $e(t)$ with its numerical approximation, we denote

$$r_n = h^{-p}(u_n - y\left(t_n\right)).$$

Then $r_0 = 0$ and for $n = 0, 1, \ldots, N-1$

$$\frac{1}{h}\left(r_{n+1} - r_n\right) = f_y\left(t_n, y\left(x_n\right)\right)r_n + \tau\left(t_n, y\left(t_n\right)\right) + O(h).$$

which is

$$\left(R_h^{\text{Euler}}\boldsymbol{r}\right)_n = \boldsymbol{\varepsilon}_n, \quad \varepsilon_n = O(h).$$

□