Introduction to Numerical Analysis I
Handout 16

1 Approximation of ODE

1.1 Euler Method

Consider an Initial Value Problem (IVP)

$$\begin{cases} y' = f(t,y) \\ y(t_0) = y_0 \end{cases} \tag{1}$$

If the equation $y' = f(t,y)$ is not separable, linear, exact, has an integration factor, Euler, Bernoulli or so, the analytical solution is not necessarily exists. Sometime the analytical solution is too complicate even if exists. In such cases it is naturally to try numerical methods to approximate its solution.

Consider an uniform grid $x_j = x_0 + jh$. Denote $y(x_n) = y_n$, then $\frac{y(x_n+h)-y(x_n)}{h} \approx \frac{y_{n+1}-y_n}{h} \approx f(x_n, y_n)$ solve it for $y_{n+1} = y_n + hf(x_n, y_n)$ to get the Euler scheme.

Error Estimation Assume that in an interval I the second derivative of the solution is bounded, $|y''| \le M$. Let L be Lipshiz constant of f in y inside a big enough rectangular $R \ni (x_0, y_0)$, that is $\left|\frac{f(x,y)-f(x,\tilde{y})}{y-\tilde{y}}\right| \le L$ for all $(x,y), (x,\tilde{y}) \in R$ (this condition could be restricted to bounded f_y). In addition define $e_n = y(x_n) - y_n$ and consider a Taylor expansion: $y(x_{n+1}) = y(x_n) + hy'(x_n) + \frac{h^2}{2}y''(c) = y(x_n) + hf(x_n, y(x_n)) + \frac{h^2}{2}y''(c)$. The error term $\frac{h^2}{2}y''(c)$ called Local Truncation Error (GTE). We next continue the global error (GTE):

$$|e_{n+1}| = |y(x_{n+1}) - y_{n+1}| =$$

$$= \left| y(x_n) - y_n + h(f(x_n, y(x_n)) - f(x_n, y_n)) + \frac{h^2}{2}y''(c) \right| \leqslant$$

$$\leqslant |y(x_n) - y_n| + h|y(x_n) - y_n| L + \frac{h^2}{2}M = (1+hL)|e_n| + \frac{h^2}{2}M$$

$$= (1+hL)|e_n| + \frac{h^2}{2}M = (1+hL)\left((1+hL)|e_{n-1}| + \frac{h^2}{2}M\right)$$

$$+\frac{h^2}{2}M = \dots (1+hL)^{n+1}|e_0| + \frac{h^2}{2}M\left(1 + (1+hL) + \dots + (1+hL)^n\right)$$

$$\underset{e_0 = y(x_0) - y_0 = 0}{=} \frac{h^2}{2}M\left(1 + (1+hL) + \dots + (1+hL)^n\right)$$

Note that $1 + hL \le e^{hL}$, since (Taylor expantion):

$$e^{hL} = e^0 + hLe^0 + \frac{(hL)^2}{2}e^c = 1 + hL + \frac{(hL)^2}{2}e^c$$

Next, using formula of sum of geometric progression one gets $e_{n+1} = \frac{h^2}{2}M\sum_{j=0}^{n}(1+hL)^j \le \frac{hM}{2L}(e^{hL(n+1)} - 1)$ Finally, since $x_{n+1} = x_0 + (n+1)h$, that is $(n+1)h = x_{n+1} - x_0$, one gets $|e_n| \le \frac{hM}{2L}(e^{L(x_n - x_0)} - 1) = O(h)$.

Note: The local error is $O(h^2)$ and the global error is $O(h)$, which is similar to integration, particularly the difference the accumulation of the error like in regular and composite rule.

1.2 Approx. of ODE using Taylor exp.

Given ODE of the form 1 Expand $y(x_{n+1})$ around x_n to get the Euler method again:

$$y(x_{n+1}) = y(x_n) + hy'(x_n) + \frac{h^2}{2}y''(c) = y_n + hf(x_n, y_n) + \text{LTE}$$

Differentiate the equation $y' = f(t,y)$ to get $y'' = f_t + f_y y' = f_t + f_y f$, which yields:

$$y_{n+1} = y_n + hf(x_n, y_n) + \frac{h^2}{2}[f_t + f_y f](x_n, y_n)$$

One may continue with the differentiation $y''' = (f_{tt} + f_{yt}f + f_y f_t) + (f_{ty} + f_{yy}f + f_y^2) f$ to get

$$y_{n+1} = y_n + hf(x_n, y_n) + \frac{h^2}{2}[f_t + f_y f](x_n, y_n) +$$

$$+\frac{h^3}{6}\left[f_{tt} + 2f_{yt}f + f_y f_t + f_{yy}f^2 + f_y^2 f\right](x_n, y_n)$$

Taking even higher derivatives of the equations is possible, but very inconvenient and also put a requirement on smoothness of f which may be very limiting.

1.3 Runge-Kutta

One may derive the following methods using Taylor expansion, but will use another approach. Given ODE of the form 1 given a grid $t_n = t_0 + nh$ integrate between the equation between t_k and t_{k+1}: That is

$$y(t_{k+1}) - y(t_k) = \int_{t_k}^{t^{k+1}} y'(t)dt = \int_{t_k}^{t^{k+1}} f(t, y(t))dt \tag{2}$$

1.3.1 Second Order Scheme

Integrating 2 using Rectangular rule yields Euler scheme: $y(t_{k+1}) = y(t_k) + (t_{k+1} - t_k)f(t_k, y(t_k)) = y_k + hf(t_k, y_k)$
Integrating 2 using Midpoint rule yields

$$y(t_{k+1}) = y(t_k) + hf\left(\tfrac{t_{k+1}+t_k}{2}, y\left(\tfrac{t_{k+1}+t_k}{2}\right)\right) =$$

$$= y(t_k) + hf\left(t_k + \tfrac{h}{2}, y\left(t_k + \tfrac{h}{2}\right)\right)$$

One changes $y\left(t_k + \frac{h}{2}\right)$ using Euler scheme $y\left(t_k + \frac{h}{2}\right) = y(t_k) + \frac{h}{2}f(t_k, y(t_k))$ to get the second order Runge-Kutta scheme, which is also called scheme of Heun

$$y(t_k + 1) = y(t_k) + hf\left(t_k + \tfrac{h}{2}, y_k + \tfrac{h}{2}f(t_k, y_k)\right)$$

On the other hand, one approximates the integral using Trapezoidal rule to get the second order Runge-Kutta scheme which is some time also called Modified Euler Scheme:

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

$$= y(t_k) + \frac{h}{2}\left(f(t_k, y(t_k)) + f(t_k + h, y_k + hf(t_k, y_k))\right))$$

The general scheme is given by

$$\begin{cases} y_{n+1} = y_n + ak_1 + bk_2 \\ k_1 = hf(x_n, y_n) \\ k_2 = hf(x_n + ch, y_n + dk_1) \end{cases}$$

The k_j's sometime called stages.

The idea is to find the coefficient a, b, c, d that maximize the accuracy or in other words minimize the error. One uses Taylor expansion of $f(t_n + ch, y_n + dk_1)$ and matches the coefficient to the Taylor expansion of y_{n+1} (like we did before), however the system is underdetermined: $a + b = 1,\ bc + bd = 1/2$. The commonly used solutions are Heun: $a = b = 1/2,\ c = d = 1$ and Modified Euler: $a = 0,\ b = 1,\ c = d = 1/2$.

1.3.2 Fourth Order Scheme

$$\begin{cases} y_{n+1} &= y_n + \frac{k_1+2k_2+2+k_3+k_4}{6} \\ k_1 &= hf(t_n, y_n) \\ k_2 &= hf\left(t_n + \frac{h}{2}, y_n + \frac{k_1}{2}\right) \\ k_3 &= hf\left(t_n + \frac{h}{2}, y_n + \frac{k_2}{2}\right) \\ k_4 &= hf(t_n + h, y_n + k_3) \end{cases}$$

One can obtain this using Simpson rule for points t_n, $t_n + \frac{h}{2}$, and t_{n+1}:

$y_(t_{n+1}) =$

$y(t_n) + \frac{h}{6}\left\{f(t_n, y(t_n)) + 4f\left(t_n + \frac{h}{2}, y\left(t_n + \frac{h}{2}\right)\right) + f(t_{n+1}, y(t_{n+1}))\right\}$

using approximation $f\left(t_n + \frac{h}{2}, y\left(t_n + \frac{h}{2}\right)\right) \approx \frac{k_2+k_3}{2}$

1.4 Multiple step methods

Definition 1.1 (Single/Multiple step scheme)**.** A single step scheme is a scheme of a general form of $y_{n+1} = g(y_n)$.

A multiple step scheme is a scheme of a general form of $y_{n+1} = g(y_n, y_{n-1}, \ldots, y_{n-p})$, for $p > 1$.

Let $p \geq 1$. We consider schemes of the following form

$$y_{n+1} = y_n + \int_{t_n}^{t_{n+1}} f(t, y(t))dt = y_n \sum_{k=n-p}^{n} a_k f(t_k, y(t_k))$$

1.4.1 Adams-Bashforth scheme

For example, require $\int\limits_{t_n}^{t_{n+1}} f(t, y(t))dt = \sum\limits_{k=n-p}^{n} a_k f(t_k, y(t_k))$ to be exact for polynomials of degree 3. Choose polynomial basis 1, $(x - x_n)$, $(x - x_n)^2$, $(x - x_n)^3$ to get

$$\sum_{k=-3}^{0} a_k = \int_{t_n}^{t_{n+1}} 1dt = t_{n+1} - t_n = h$$

$$\sum_{k=-3}^{0} (kh)a_k = \sum_{k=n-3}^{0} (t_k - t_n)a_k = \int_{t_n}^{t_{n+1}} (t - t_n)dt = \frac{h^2}{2}$$

$$\sum_{k=-3}^{0} (kh)^2 a_k = \sum_{k=n-3}^{0} (t_k - t_n)^2 a_k = \int_{t_n}^{t_{n+1}} (t - t_n)^2 dt = \frac{h^3}{3}$$

$$\sum_{k=-3}^{0} (kh)^3 a_k = \sum_{k=n-3}^{0} (t_k - t_n)^3 a_k = \int_{t_n}^{t_{n+1}} (t - t_n)^3 dt = \frac{h^4}{4}$$

The system above has a solution $a_{-3} = -\frac{9}{24}h$, $a_{-2} = \frac{37}{24}h$, $a_{-1} = -\frac{59}{24}h$, $a_0 = \frac{55}{24}h$. The local truncation error (LTE) of this scheme is $O(h^5)$.

1.4.2 Leap-Frog Scheme

Another multiple step scheme can be constructed using midpoint rule as following:

$$y_{k+1} - y_{k-1} = \int_{t_{n-1}}^{t_{n+1}} f(t, y(t))dt = 2hf(t_k, y(t_k))$$

That is we got Leap-Frog scheme $y_{k+1} = y_{k-1} + 2hf(t_k, t_k)$ which has LTE of $O(h^2)$. This scheme require 2 initial guesses. One is obtained from the initial condition of the equation, while another is missing. To use schemes like this one uses other schemes until until enough initial guesses are obtained.

1.5 Implicit Scheme

A scheme is called Implicit if y_{n+1} appears on two sided of the equations. For example, by using reverse rectangular rule in integration one get **Backward Euler Scheme** $y_{n+1} = y_n + hf(x_{n+1}, y_{n+1})$. In order to use scheme like this one need to solve the implicit equation, which is usually not simple

Crank-Nucholson When we used trapezoidal rule we got $y_{n+1} = y_n + \frac{h}{2}\left(f(x_n, y_n) + f(x_{n+1}, y_{n+1})\right)$ This scheme is called **Crank-Nucholson**.

1.6 Consistence

We will learn one interesting method to analyze the convergence of numerical scheme to the solution of ODE.
Reminder: Characteristic equation of an ODE of the form $a_0 y + a_1 y' + a_2 y'' + \cdots + a_n y^{(n)} = 0$ is obtained using an assumption (an educated guess) that $y(t) = e^{rt}$, substituting it into: $e^{rt} = a_0 + a_1 r + a_2 r^2 + \cdots + a_n r^n = 0$ and finally simplify to get $a_0 + a_1 r + a_2 r^2 + \cdots + a_n r^n = 0$. That is the roots of the characteristic equation, $r_0, \ldots, r_n$ gives the solution to the ODE: $y(t) = e^{r_0 t} + \cdots + e^{r_n t}$.

Consider now a scheme of a general form $y_{n+1} = \sum\limits_{k=n-p}^{n} a_k y_k$, where $p > 1$. The characteristic polynomial/equation of the scheme. Similar to characteristic equation of the ODE we define characteristic equation using the transformation $y_k \mapsto r^k$ to get the characteristic polynomial $r^{n+1} = \sum\limits_{k=n-p}^{n} a_k r^k$ rewrite as

$$0 = r^{n+1} - \sum_{k=n-p}^{n} a_k r^k = r^{n-p}\left(r^{p+1} - \sum_{k=0}^{p} a_k r^k\right).$$

Let $r_0, \ldots, r_{p+1}$ roots of characteristic polynomials, then $y_{n+1} = \sum\limits_{k=0}^{p+1} c_k r_k^{n-p} = 0$. More precisely the roots r_k's are functions of h, that is $r_k = r_k(h)$.

Definition 1.2. We say that the multiple-step scheme is always converges if the (simple) roots of characteristic equations $r_k(h)$, $k = 0, ..., p + 1$ satisfy either:

- Is a parasitic term that tends to zero: $\lim\limits_{h \to 0} r_k^{n-p} = 0$, or
- Is a principle root that tends to the solution: $\lim\limits_{h \to 0} r_k^{n-p} =$ solution.

Otherwise, the scheme may not converge.