Numerical Methods for the IVP

$$u'(t) = f(u) \qquad t_n = t_0 + nk$$
$$u(t_0) = \eta \qquad U^n \approx u(t_n)$$

Basic methods:

① Explicit Euler

$$\frac{U^{n+1} - U^n}{k} = f(U^n)$$

$$U^{n+1} = U^n + kf(U^n)$$

② Implicit Euler

$$\frac{U^{n+1} - U^n}{k} = f(U^{n+1})$$

Requires solving a system of equations at each step.

Trapezoidal method

$$\frac{U^{n+1}-U^n}{k}=\frac{1}{2}\left(f(U^n)+f(U^{n+1})\right)$$

Leapfrog (multistep)

$$\frac{U^{n+1}-U^{n-1}}{2k}=f(U^n)$$

Local truncation error

$$\tau^n=\frac{u(t_{n+1})-u(t_n)}{k}-\frac{1}{2}\Big(\underbrace{f(u(t_n))}_{u'(t_n)}+\underbrace{f(u(t_{n+1}))}_{u'(t_{n+1})}\Big)$$

$$u(t_{n+1})=u(t_n)+ku'(t_n)+\frac{k^2}{2}u''(t_n)+\frac{k^3}{6}u'''(t_n)+O(k^4)$$

$$f(u(t_{n+1}))=u'(t_n)+ku''(t_n)+\frac{k^2}{2}u'''(t_n)+\frac{k^3}{6}u''''(t_n)+O(k^4)$$

$$\tau^n=\frac{\cancel{u}+\cancel{ku'}+\cancel{\frac{k^2}{2}u''}+\frac{k^3}{6}u'''-\cancel{u}}{k}$$

$$-\frac{1}{2}\left(\cancel{u'}+\cancel{u'}+\cancel{ku''}+\frac{k^2}{2}u'''\right)+O(k^3)$$

$$\tau^n=\frac{k^2}{6}u'''-\frac{k^2}{4}u'''+O(k^3)=-\frac{k^2}{12}u'''+O(k^3)$$

2nd-order accurate.

If we write

$$U^{n+1} = U^n + \frac{k}{2}\left(f(U^n) + f(U^{n+1})\right)$$

4)

$$\mathcal{L}^n = u(t_{n+1}) - u(t_n) - \frac{k}{2}\left(f(u(t_n)) + f(u(t_{n+1}))\right)$$

$$\mathcal{L}^n = k\tau^n$$

one-step error $\quad$ local truncation error

$u \in \mathbb{R}^m$

$f(u): \mathbb{R}^m \to \mathbb{R}^m$

How to achieve higher order?

1. Use more derivatives of u.

$$u(t_{n+1}) \approx u(t_n) + k u'(t_n) + \frac{k^2}{2} u''(t_n)$$

$$U^{n+1} = U^n + k f(U^n) + \frac{k^2}{2} f'(U^n) f(U^n)$$

2nd-order accurate

$$\frac{d^2}{dt^2} u(t) = \frac{d}{dt} f(u(t)) = f'(u)\, u'(t) = f'(u) f(u)$$

$$\frac{d^2 u_i}{dt^2} = \sum_j \frac{\partial f_i}{\partial u_j} f_j(u)$$

These are called <u>Taylor series</u> methods.

$$\frac{d^3 u_i}{dt^3} = \frac{d}{dt}\sum_j \frac{\partial f_i}{\partial u_j} f_j(u)$$

$$= \sum_j \frac{\partial f_i}{\partial u_j} \sum_k \frac{\partial f_j}{\partial u_k} \overset{f_k(u)}{\frac{du_k}{dt}} + \sum_j f(u)_j \sum_k \frac{\partial^2 f_i}{\partial u_j \partial u_k} \overset{f_k(u)}{\frac{du_k}{dt}}$$

$$= f'(f'(f(u))) + f''(f(u), f(u))$$

② Use more evaluations of f

Example: $U^* = U^n + \frac{k}{2} f(U^n)$

(midpoint Runge-Kutta method) $U^{n+1} = U^n + k f(U^*)$

$$\frac{u(t_{n+1}) - u(t_n)}{k} = \frac{\cancel{u} + ku' + \frac{k^2}{2}u'' + \frac{k^3}{6}u''' + O(k^4) - \cancel{u}}{k}$$

$$u' + \frac{k}{2}u'' + \frac{k^2}{6}u''' + O(k^4)$$

$$U^{n+1} = U^n + kf\left(U^n + \frac{k}{2}f(U^n)\right)$$

$$f\left(U^n + \frac{k}{2}f(U^n)\right) = f(U^n) + \frac{k}{2}f'(U^n)f(U^n) + \frac{k^2}{8}\left(f''(f(U^n), f(U^n)) + f'(f'(f(U^n)))\right) + O(k^3)$$

$$f'(f(u))$$

$$\tau^n = \frac{u(t_{n+1}) - u(t_n)}{k} - \underline{f(u(t_n))} - \underline{\frac{k}{2}f'(f(u(t_n)))} - \frac{k^2}{8}\left(f''(f(u(t_n)), f(u(t_n))) - f'(f'(f(u(t_n))))\right) + O(k^3)$$

$$\tau^n = \cancel{u} + \cancel{\frac{k}{2}u'} + \frac{k^2}{6}u''' - \cancel{u} - \cancel{\frac{k}{2}u'} - \frac{k^2}{8}\underbrace{\left(f''(f,f) + f'(f'(f))\right)}_{u'''(t_n)} + O(k^3)$$

$$\tau^n = \frac{k^2}{24}u'''(t_n) + O(k^3)$$

2nd-order accurate

Advantages of Runge-Kutta:

- self-starting
- only need f, no other derivatives
- easy to adapt k

if $f(u) = Lu$

$f'(u) = L$

$f''(u) = 0$

③ Use more previous steps

Examples: (Leapfrog)

$$\frac{U^{n+1} - U^{n-1}}{2K} = f(U^n)$$

Next time

MATLAB: ode113(s)

- Not self-starting
- Trickier to adapt K
- Only need 1 evaluation of $f(u)$ per step.

Non-adaptive: $t_n = t_0 + nK$

Adaptive: $t_n = t_{n-1} + K_n$

K_n not necessarily equal to K_{n-1}