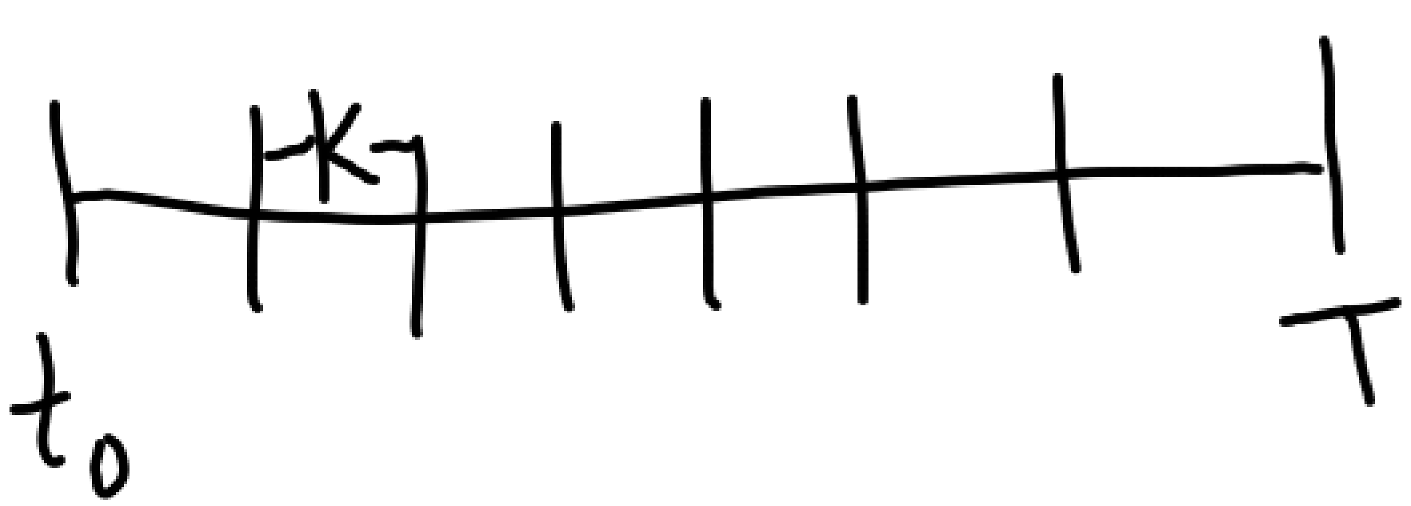
Numerical Methods for the initial-value problem

$$u'(t) = f(u) \qquad u : \mathbb{R} \to \mathbb{R}^N$$

$$u(t_0) = \eta$$

Discretize:

$$t_n = t_0 + nk$$



$$U^n \approx u(t_n)$$

Unlike the BVP, we can "march" along the grid, starting at t_0.

Basic methods

① Explicit Euler

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

1st-order accurate

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

Explicit (nothing to solve)

② Implicit Euler

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

Need to solve equations at each step.

More expensive (per step) but more stable.

---

③ Trapezoidal method

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

2nd-order accurate

Local Truncation Error

Replace $U^n \to u(t_n)$ in our method:

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

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

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

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

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

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

One-step error

This is what we obtain if we start by writing the method in the form

$$U^{n+1} = \ldots$$

For trapezoidal:

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

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

For the trapezoidal method, $\mathcal{L}^n = \mathcal{O}(k^3)$

At each step we make an error of $\mathcal{O}(k^3)$.

But we take $\mathcal{O}(k^{-1})$ steps.

So we expect an error of $\mathcal{O}(k^3 k^{-1}) = \mathcal{O}(k^2)$ at time T.

How to achieve higher order?

① Taylor series methods (use more derivatives of u)

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

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

$$u''(t) = \frac{d}{dt} f(u(t))_i = \sum_j \frac{\partial f_i}{\partial u_j} \frac{du_j}{dt} = \sum_j \frac{\partial f_i}{\partial u_j} f_j$$

$$= f' f$$

Difficulty: We have to compute derivatives of f.

$$\frac{d^3 U(t)_i}{dt^3} = \frac{d}{dt}\sum_j \frac{\partial f_i}{\partial u_j} f_j$$

$$= \sum_j \frac{\partial f_i}{\partial u_j}\frac{d f_j}{dt} + \sum_j f_j \frac{d}{dt}\frac{\partial f_i}{\partial u_j}$$

$$= \sum_j \frac{\partial f_i}{\partial u_j}\sum_k \frac{\partial f_j}{\partial u_k} f_k + \sum_{j,k} f_j \frac{\partial^2 f_i}{\partial u_j \partial u_k} f_k$$

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

The number of terms grows exponentially with the order.

② Use more evaluations of f (Runge-Kutta methods)

For example:

Midpoint RK

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

$$U^{n+1} = U^n + k f(U^*)$$

$$U^{n+1} - U^n - k f\left(U^n + \frac{k}{2} f(U^n)\right) = 0$$

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

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

We get

$$\underline{kU'(t_n)} + \underline{\frac{k^2}{2} U''(t_n)} + \frac{k^3}{6} U'''(t_n) - \left(\underline{kU'(t_n)} + \underline{\frac{k^2}{2} f'f} + \frac{k^3}{8} f''(f,f)\right) + O(k^4) = k\tau^n$$

$$\tau^n = k^2\left(\frac{1}{6} U'''(t_n) - \frac{1}{8} f''(f,f)\right) + O(k^3)$$

The method is 2nd-order accurate.

Advantages of RK methods:

- Only need f, not further derivatives
- Self-starting (only need η to begin)
- Easy to change/adapt step size k.

MATLAB: ode45

RK method with 5 evaluations of f per step.

③ Use more previous step values of U, f. (Linear multistep methods)

Example: Leapfrog

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

Advantage:
- Only need 1 evaluation of f per step

Disadvantages:
- Need multiple values to begin
- "Tricky" to adapt step size K

MATLAB: ode113
Multistep methods of 1 to 13 steps