Runge-Kutta Method

Next, we only consider the system is scalar ODE since we can extend it to high dimension by applying component-wise. And denote y_k is time instance

consider IVP: $y'(t) = f(t, y(t))$, $t_0 \le t \le T$.
$y(t_0) = y_0$.

(1) Euler method:

① Explicit Euler method:

Take one step: $y_1 = y_0 + h f(t_0, y_0)$. i.e. choose the slope at starting point.

Algorithm: $h = (T - t_0)/N$. $t_{k+1} = t_k + h$.

$$y_{k+1} = y_k + h f(t_k, y_k).$$

② Implicit Euler method:

Take one step: $y_1 = y_0 + h f(t_1, y_1)$. i.e. the slope is of ending point. then

We also need to solve out y_1.

Algorithm: $h = (T-t_0)/N$. $t_{k+1} = t_k + h$.

$$y_{k+1} = y_k + f(t_k+h, y_{k+1})$$

Rmk: i) Choose which method also depend on whether it's easy to solve y_{k+1}

ii) We can apply Newton's method to find y_{k+1} as root of $y_{k+1} - y_k - f(\square) = 0$

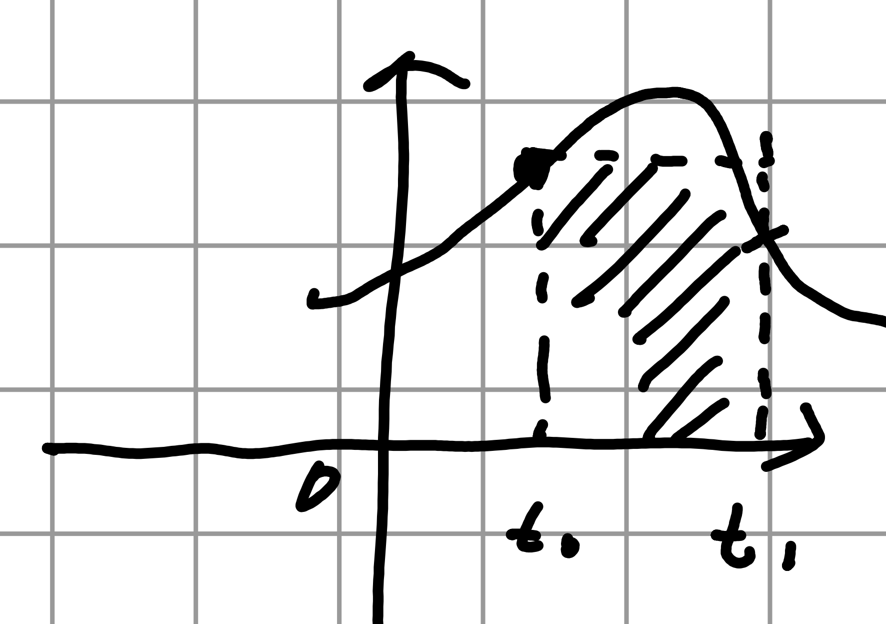
$\rightsquigarrow$ <u>Runge-Kutta Scheme:</u>

Note IVP $\Leftrightarrow$ $y(t_1) = y_0 + \int_{t_0}^{t_1} f(t, y(t))\,dt$

So: the idea is to evaluate the integral

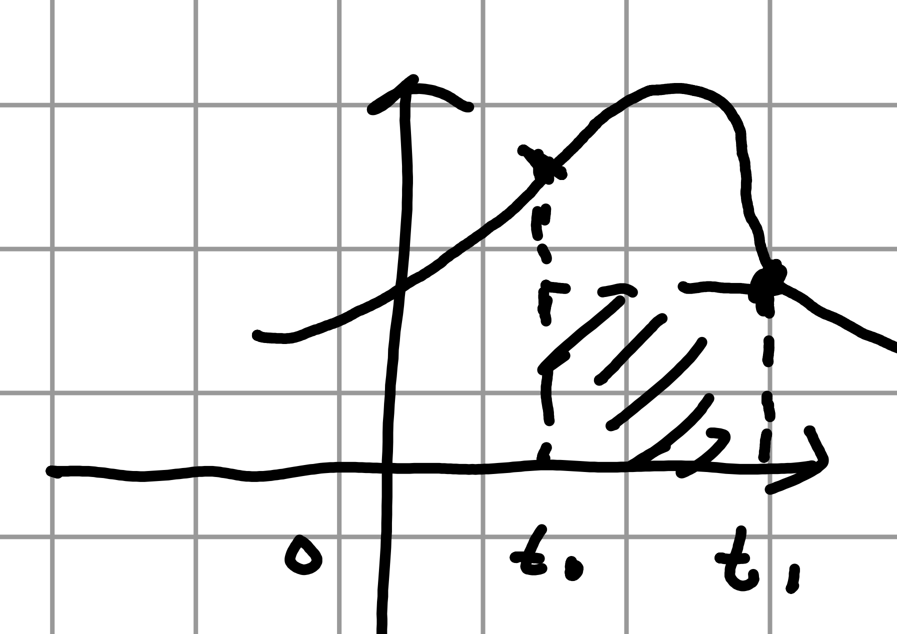
<u>e.g.</u> i) Explicit Euler:

replace $f(t, y(t))$ by $f(t_0, y_0)$ for integrand.



ii) Implicit Euler:

replace $f(t, y(t))$ by $f(t_1, y(t_1))$



Rmk: The area ($\int_{t_0}^{t_1} f(t, y(t))\,dt$) is quite different with !

iii) (Implicit Trapezoidal)

$$y_1 = y_0 + \frac{h}{2}\left[f(t_0, y_0) + f(t_1, y_1) \right].$$

iv) (Heun / explicit Trapezoidal)

To modify above, we replace y_1 by

$\hat{y}_1 = y_0 + h f(t_0, y_0)$ with

$$y_1 = y_0 + \frac{h}{2}\left[f(t_0, y_0) + f(t_1, \hat{y}_1) \right].$$

$\Rightarrow$ Consider general quadrature formula on $[0,1]$ with s nodes $\{\hat{c}_i\}_{i=1}^s$. (pt. to be evaluated) and corresp'd weights $\{\hat{w}_i\}_{i=1}^s$: $\int_0^1 g(s)\,ds \approx \sum_1^s g(\hat{c}_i)\hat{w}_i$.

Apply on the IVP. rewrite $\int_{t_0}^{t_1} f(s, y(s))\,ds$

$= \int_0^1 h f(t_0 + sh, y(t_0 + sh))\,ds.$

$\approx \sum_1^s h\hat{w}_i f(t_0 + \hat{c}_i h, y(t_0 + \hat{c}_i h)).$

<u>Def</u>: For $b_i, a_{ij} \in \mathbb{R}$, $i,j = 1, \dots, r$. $c_i \triangleq \sum_{j=1}^r a_{ij}$

$$k_i = f\left(t_n + c_i h, y_n + h\sum_{j=1}^r a_{ij} k_j\right), \quad i = 1, \dots, r.$$

$y_{n+1} = y_n + h \sum_1^r b_i k_i$. defines a RK one-step method with r stages for solving IVP.

Rmk: Write in Butcher-tableau:

$$\begin{array}{c|c} \underline{c} & \underline{A} \\ \hline & \underline{B} \end{array} = \begin{array}{c|ccc} c_1 & a_{11} & \cdots & a_{1r} \\ \vdots & \vdots & & \\ c_r & a_{r1} & \cdots & a_{rr} \\ \hline & b_1 & \cdots & b_r \end{array}$$

e.g. i) explicit RK scheme:

$$\begin{array}{c|cccc} c_1 & 0 & & & 0 \\ \vdots & a_{21} & \ddots & & \\ & \vdots & & \ddots & \\ c_r & a_{r1} & \cdots & a_{r,r-1} & 0 \\ \hline & b_1 & \cdots & & b_r \end{array}$$

Rmk: if diagonal elements are not $= 0$. then we call it DIRK. i.e. diagonal-implicit RK method.

ii) explicit Euler: $r=1$ $\begin{array}{c|c} 0 & 0 \\ \hline & 1 \end{array}$

iii) implicit Euler: $r=1$ $\begin{array}{c|c} 1 & 1 \\ \hline & 1 \end{array}$

iv) classical RK / RK4: $\begin{array}{c|cccc} 0 & 0 & & & \\ 1/2 & 1/2 & 0 & & \\ 1/2 & & 1/2 & 0 & \\ 1 & 0 & 0 & 1 & 0 \\ \hline & 1/6 & 1/3 & 1/3 & 1/6 \end{array}$

Rmk: It's generalization of the Euler method. $\{b_i\}$. $\{k_i\}$ are like some weights.

(3) **Discretization error:**

Def: i) Global discretization error is $e_h(t_n) = y(t_n) - y_n$. $n = 1, \cdots N$. (y_n is n^{th} est.)

Rmk: $e_h(t_n)$ is error accumulated until $t = t_n$. And we care

i) $\| e_h(T) \| = \| y(T) - y_N \|$

ii) $\max_{k=1,\cdots N} \| e_h(t_k) \|$.

ii) A method is called convergent of order p if $\max_{0 \le k \le N} \| e_h(t_k) \| = O(h^p)$.

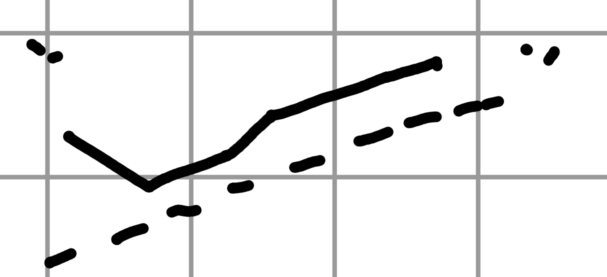
Rmk: i) Euler method $\sim O(h)$

Heun $\sim O(h^2)$.

RK4 $\sim O(h^4)$.

ii) We can also run the convergence test(*) to find them

(*) Order depend on slope.

by using loglog plot. Sometimes inconsistency  in plot occurs which is from computational error.

iii) One step error / local discret. error for a RK scheme is $\underline{\delta}_{n+1} = \underline{y}(t_{n+1}) - \tilde{\underline{y}}_{n+1}$.

where $\tilde{\underline{y}}_{n+1} = \underline{y}(t_n) + h \sum_{1}^{s} b_i \underline{k}_i$ and $\underline{k}_i = \underline{f}(t_n + c_i h, \underline{y}(t_n) + h \sum_{j}^{s} a_{ij} \underline{k}_j)$. $i = 1, 2, \dots, s$.

Rmk: We use the exact value $\underline{y}(t_n)$ rather $\underline{y}_n$ because we only care error of one-step and avoid accumulations of error.

e.g. (explicit Euler)

$|\underline{e}_h(t_N)| = O(h)$, $|\underline{\delta}_1| = O(h^2)$.

Rmk: $\underline{e}_h(t_1) = \underline{\delta}_1$ at 1^{st} step.

iv) Truncation error is: $\underline{\tau}_{n+1} = \underline{\delta}_{n+1}/h$

$$= \frac{\underline{y}(t_{n+1}) - \underline{y}(t_n)}{h} - \sum_{1}^{s} b_i \underline{k}_i .$$

Rmk: $y(t_n)$ is used to compute k_i

ii) A RK method is consistent if

$\max_{0\le k\le N} \| z_k \| \xrightarrow{h\to 0} 0$. it's of order p if

$\max_{k} \| z_k \| = O(h^p)$

Rmk: Other references might use different defs for local error / one steperror / truncation error. The difference is whether to divide h.

e.g. Explicit Euler is consistent of order $O(h)$.

Rmk: We use one-step method above:

i.e. $y_0 \to y_1 \to \cdots \to y_k \to \cdots$

There's also linear multilinear method

i.e. $y_0 \to y_1 \to y_2 \to y_3 \cdots$

"RK scheme" belongs to one-step method.

(4) Consistency order for RK:

Next, we consider f is smooth. $y(t) \in \mathbb{R}^1$.

<u>Rmk</u>: If f isn't regular enough, then the convergence may fail.

1°) Apply Taylor expansion on $y(t)$:

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

$y^{(2)}(t) = \frac{d}{dt} f(t, y(t)) = (f_t + f_x f)(t, y(t))$.

S_1: $y(t) = y(t_k) + f(t_k, y(t_k)) h + \frac{1}{2} h^2 (f_t + f_x f)(t_k, y(t_k)) + O(h^3)$.

where $h = t - t_k$.

<u>e.g.</u> (Explicit Euler: Consistency order 1)

From above. Let $t = t_{k+1}$. We get

$\zeta_{k+1} = \frac{1}{2} h^2 y^{(2)}(t_k) + O(h^3)$. Where

$y^{(2)}(t_k) \neq 0$ generally (That's why we compute $y^{(2)}(t)$).

S_1: $\max_k |\zeta_{k+1}| \leq \frac{1}{2} h^2 \max_{[t_0, T]} |y^{(2)}(t)|$.

2°) target: choose a_{ij}, b_i, c_i s.t. it has p-order consistency.

Note $\tau_{n+1} = \frac{y(t_{n+1}) - y(t_n)}{h} - \sum_1^s b_i k_i$.

i) To be consistent ($\tau_n(h) \xrightarrow{h \to 0} 0$):

Recall Taylor expansion on $y(t)$. We need: $\sum b_i k_i = f(t_n, y(t_n)) + O(h)$.

Expand $k_i = f(t_n, y(t_n)) + f_t(t_n, y(t_n)) c_i h$
$+ f_x(t_n, y(t_n)) h \sum_{j=1}^s a_{ij} k_j + O(h^2)$.

$\Longleftrightarrow \sum_1^s b_i = 1$

ii) To be consistent of order two:

We need: $\sum b_i k_i = f(t_n, y(t_n)) + \frac{1}{2} h \cdot$
$(f_t + f_x f)(t_n, y(t_n)) + O(h^2)$.

e.g. Explicit method with 2 stages:

$s = 2$, $a_{11} = a_{12} = a_{22} = 0$.

Plug the expansions of k_i in the equation above. We have:

$b_1 + b_2 = 1$. $c_2 b_2 = \frac{1}{2}$. $a_{21} b_1 = \frac{1}{2}$

Rmk: One choice of a_{ij}, c_i, b_i is Heun's method.

Butcher barriers for explicit RK method:

converge order p	1	2	3	4	5	6	7
min # stages	1	2	3	4	6	7	9

Rmk: $p \le 5$.

Why high order?

Assume: i) $[t_0, T] = [0, 1]$.

ii) Step length $h \Rightarrow \frac{1}{h}$ times step

iii) Each step error of size h^{p+1}. Old error won't amplify but only add up.

iv) Computer isn't exact, having some error up to 10^{-m}.

$\Rightarrow$ error $\approx$ # step $\times$ error in each step

$$\approx \frac{1}{h}(h^{p+1} + 10^{-m}) = h^p + 10^{-m}/h.$$

We get the min error $\approx O(10^{-mp/p+1})$

Rmk: For explicit Euler $\sim O(10^{-8})$.

For RK4 $\sim O(10^{-13})$.

(5) Connection of consistency and convergence:

① Simplified idea:

Assume i) one-step error $\approx O(h^{p+1})$.

ii) old error doesn't increase, only add up

$\Rightarrow$ global error $\sim N \cdot Ch^{p+1} = \frac{T-t_0}{h} \cdot Ch^{p+1}$

$\sim O(h^p)$.

Rmk: We lose one-order error compared to one-step error.

For it holds in general. We need some kind of "Stability".

e.g. (Explicit Euler)

Assume $f(t,y)$ is conti. on $I \times \mathbb{R}^1$ and globally Lip at y-variable.

Also, we require f smooth enough:

Recall $z_{n+1} = \xi_{n+1}/h = \dfrac{y(t_{n+1}) - y(t_n) - hf(t_n, y(t_n))}{h}$

$\sim O(h)$ and $y_{n+1} = y_n + hf(t_n, y_n)$

We have: $e(t_{n+1}) = y(t_{n+1}) - y_{n+1} =$

$$y(t_n) + hf(t_n, y(t_n)) + hz_{n+1} - y_n - hf(t_n, y_n)$$

$$= e(t_n) + h(f(t_n, y(t_n)) - f(t_n, y_n)) + hz_{n+1}$$

Apply Lip-conti. of f. We get:

$|e(t_{n+1})| \leq |e(t_n)| + hL|e(t_n)| + h|z_{n+1}|$.

<u>Rmk</u>: $(1+hL)|e(t_n)|$ is amplification of old error and $h|z_{n+1}|$ is new error.

So: $|e(t_n)| \leq \sum_0^{n-1} Lh|e(t_k)| + (|e(t_0)| + \sum_1^n h|z_k|)$

<u>Lem.</u> (Discrete Gronwall's inequi.)

$(w_n), (a_n), (b_n) \subseteq \mathbb{R}^{\geq 0}$. St. $w_0 = b_0$. &

$w_n \leq \sum_{k=0}^{n-1} a_k w_k + b_n$, $n \geq 1$. If $(b_k) \nearrow$,

Then: $w_n \leq b_n \exp(\sum_0^{n-1} a_k)$. $\forall n \geq 1$.

<u>Pf</u>: Set $\lambda_0 = b_0 - w_0$. $\lambda_n = \sum_0^{n-1} a_k w_k + b_n - w_n$

Set $S_n = W_n + h_n$. Next we prove:

$S_n \le b_n \exp(\sum_0^{n-1} a_k)$ by induction.

$n=0$: $S_0 = W_0 + h_0 = b_0 \le \exp(0) b_0$.

Note
$$S_n = S_{n-1} + a_{n-1} W_{n-1} + b_n - b_{n-1}$$
$$\overset{W\le S}{\le} (1+a_{n-1}) S_{n-1} + b_n - b_{n-1}$$
$$\overset{hypo}{\le} (1+a_{n-1}) b_{n-1} e^{\sum_0^{n-2} a_k} + (b_n - b_{n-1})$$
$$\overset{e^x \ge 1+x}{\le} b_{n-1} e^{a_{n-1}} e^{\sum_0^{n-2} a_k} + e^{\sum_0^{n-1} a_k} (b_n - b_{n-1})$$
$$= b_n \exp(\sum_0^{n-1} a_k).$$

So we have: $|e(t_N)| \le (|e(t_0)| + \sum_1^N h|\tau_k|) e^{\sum Lh}$

$$= e^{L(T-t_0)} (|e(t_0)| + \sum_1^N h|\tau_k|)$$

$$\Rightarrow \max_{1\le n\le N} |e(t_n)| \le e^{L(T-t_0)} (|e(t_0)| + (T-t_0) \max_k |\tau_k|)$$

Rmk: We can control $|e(t_0)|$ s.t. it decays fast or $|e(t_0)| = 0$. So we have:

Consistent of order p $\Rightarrow$ convergent of order p.

② General Case:

Assume $f(t,y)$ is conti. on $I \times \mathbb{R}^1$ and Lip.

Consider general case: $h_k \triangleq t_k - t_{k-1}$ and $h := \max_{1 \le k \le N} h_k$, where $t_N = T$. Consider general one-step method: $y_n = y_{n-1} + h_n F(h_n; t_{n-1}, y_n, y_{n-1})$

Def: i) $(L_h y^h)_n = \dfrac{y_n - y_{n-1}}{h_n} - F(h_n; t_{n-1}, y_n, y_{n-1})$

Rmk: So truncated error $\tau_n = (L_h y)_n$

$= \dfrac{y(t_n) - y(t_{n-1})}{h_n} - F(h_n; t_{n-1}; y(t_n), y(t_{n-1}))$

ii) An update formula for a one-step method is called Lip-conti. if $\exists L > 0$ s.t. $|F(h, t, x, y) - F(h, t, \tilde{x}, \tilde{y})| \le L \cdot (|x - \tilde{x}| + |y - \tilde{y}|)$, $\forall t, x, \tilde{x}, y, \tilde{y}$

Rmk: In RK-method, f is Lip-conti.

$\Rightarrow$ update formula is Lip-conti.

(proved by inducting on s)

Thm. (Discrete Stability)

Lip-conti. update formula is discretely stable, i.e. $\forall y^{(h)} := \{y_n\}$, $z^{(h)} := \{z_n\}$ grid function on $\mathbb{N}$. If $h < \frac{1}{2} L^{-1}$. Then:

$$|y_n - z_n| \leq e^{kL(t_n - t_0)} \Big(|y_0 - z_0| + \sum_1^n h_k |(L_h y^h - L_h z^h)_k| \Big)$$

where const. $k = 4$ for implicit method &

$k = 1$ for explicit method (then no need

for condition "$h < \frac{1}{2} L^{-1}$").

<u>Pf</u>: By def of $L_h y^h$, $L_h z^h$. We have:

$$y_n - z_n = y_{n-1} - z_{n-1} + h_n \big(F(h_n, t_{n-1}, y_n, y_{n-1}) - F(h_n, t_{n-1}, z_n, z_{n-1}) \big) + h_n (L_h y^h - L_h z^h)_n.$$

Set $e_n = y_n - z_n$, $\Sigma_n = (L_h y^h - L_h z^h)_n$.

1) Explicit case:

$$|e_n| \overset{Lip}{\leq} |e_{n-1}| + h_n L |e_{n-1}| + h_n |\Sigma_n|.$$

So: $|e_n| \leq \sum_0^{n-1} h_k L |e_k| + |e_0| + \sum_1^n h_k |\Sigma_k|$.

Apply discrete Grownwall's:

$$|e_n| \leq e^{L(t_n - t_0)} \Big(|e_0| + \sum_1^n h_k |\Sigma_k| \Big)$$

2) Implicit case:

$$|e_n| \overset{Lip}{\leq} |e_{n-1}| + h_n L (|e_{n-1}| + |e_n|) + h_n |\Sigma_n|$$

Set $v_n = (1 - h_n L)|e_n|$. Under the

assumption $h < \frac{1}{2} L^{-1}$. We have:

$$W_n \le W_{n-1} + \frac{h_n + h_{n-1}}{1 - h_{n-1} L} W_{n-1} + h_n |\varepsilon_n|.$$

So: $W_n \le \sum_{0}^{n-1} \frac{h_{k+1} + h_k}{1 - h_k L} L W_k + W_0 + \sum_{1}^{n} h_k |\varepsilon_k|$

Apply discrete Gronwall's:

$$W_n \le (W_0 + \sum_{1}^{n} h_k |\varepsilon_k|) e^{L \sum_{0}^{n-1} (h_{k+1} + h_k)/1 - h_k L}$$

$$\Rightarrow |e_n| \le \frac{1}{1 - h_n L} \cdot \square \quad (\text{By } \frac{1}{1-x} \le e^{\frac{x}{1-x}})$$

$$\le e^{\frac{h_n L}{1 - h_n L}} e^{L \Sigma \square} \cdot (W_0 + \sum_{1}^{n} h_k |\varepsilon_k|)$$

Note $1 - h_k L \ge 1 - hL \ge \frac{1}{2}$.

So: $|e_n| \le e^{4 \Sigma h_k} \cdot (W_0 + \sum_{1}^{n} h_k |\varepsilon_k|)$

$$= e^{4(t_n - t_0)} (W_0 + \sum_{1}^{n} h_k |\varepsilon_k|).$$

Cor. For update formula is Lip-conti. and consistent. If $|y_0^h - y(t_0)| \to 0$

Then: $\max_{t_n} |y(t_n) - y_n| \to 0$. $(h \to 0)$

Besides, $|y(t_n) - y_n| \le e^{KL(t_n - t_0)} \cdot (|y_0^h - y(t_0)| + \sum_{1}^{n} h_k |\varepsilon_k|)$ holds as above $0 \le n \le N$.

Pf: Set: $y^h = (y_n)$. compute sol.

So: $L_h(y^h)_n = 0.\ \forall n.$

And let $z^h = (y(t_n))_n$ exact solution

So: $L_h(z^h)_n = \tau_n. \Rightarrow \varepsilon_n = \tau_n.$

<u>Rmk</u>: It implies that under the stability condition. we have:

p-order consistent $\Rightarrow$ p-order converge.

(In most of cases, we can let $y_0 = y(t_0)$ and $\|y_0 - y(t_0)\| = 0$.

Sometimes we need to prior one $\tilde{y}_0$ estimate. And $\|\tilde{y}_0 - y(t_0)\|$ is neglectable)

(b) <u>Adaptive time stepping for RK</u>:

For uniform step length h:

<u>Pro</u>: it's easy.

<u>Cons</u>: i) How to choose proper h?

ii) Insuff. to use same step length everywhere: <u>e.g.</u>

iii) It won't detect the exact position of singularities.

Rmk: Because we don't know how the exact solution look like.

$\Rightarrow$ Introduce "adaptive step length control".

Goal: i) Use as few steps as possible.

ii) Try to enforce a given tolerance.

e.g. $TOL \geq \| y(T) - y_N \|$)

Ingredients:

i) Error estimator ii) Estimate of new step length.

iii) Computation of local TOL_k.

Rmk: Error is from i) new step error and ii) Amplification of old error.

Next, we only focus on the error i):

Con: No guarantee for accuracy at T.

pros: a) Cheap b) typically works

c) No need to reject previous steps.

(i.e. error large $\not\Rightarrow$ start again)

Approach: Control one-step error by adjusting the step length $h_{k+1} = t_{k+1} - t_k$.

Rmk: And we need to estimate the local one-step error.

Algorithm:

Given t_k, y_k and suggestion for step length $h_{k+1} = t_{k+1} - t_k$. We want to get y_{k+1}:

i) Compute local tolerance tol_{k+1} on $[t_k, t_{k+1}]$

ii) Take step $y_k \to y_{k+1}$ by estimating error:

a) Error $\leq tol_{k+1}$ $\to$ accept y_{k+1}

b) Error $> tol_{k+1}$ $\to$ repeat with $\frac{h_{k+1}}{2}$.

① Computation of TOL_{k+1}:

e.g. I) Goal: local error is propositional to step length h_{k+1}: $tol_{k+1} := \frac{h_{k+1}}{T-t_0} TOL$.

If no amplification of previous errors.

$$\text{Total error} = \sum \text{local error} \leq \sum \frac{h_k}{T-t_0} TOL = TOL.$$

ii) Provide relative/absolute tolerance RelTOL & AbsTOL. For example:

$tol_{k+1} = RelTOL \cdot \|y\| \vee AbsTOL.$

(2) Estimate local error as posterior:

take a step and use the result to estimate how big the error was in this step.

Strategies: i) Step length halving:

use method of order p. For given step length h:

$t_k \xrightarrow{h} t_{k+1}$. $y_k \xrightarrow{h} y^h_{k+1}$. Similarly for reaching y_{k+1}: $y_k \xrightarrow{h/2} \tilde{y} \xrightarrow{h/2} y^{\frac{h}{2}}_{k+1}$.

$EST_{k+1} = |y^h_{k+1} - y^{\frac{h}{2}}_{k+1}| / (1-2^{-p})$.

Rmk: 2^{-p} is the scaling factor from $O(h^p) \to O((h/2)^p) = 2^{-p} O(h^p)$. And we want to erase its effect.

ii) Method of different orders:

take time step h_{k+1} with two method of different orders:

Start from (t_k, y_k):

$y_{k+1}(y_k)$ is approxi. sol. at t_{k+1} of lower order.

$\tilde{y}_{k+1}(y_k)$ is approxi. sol. at t_{k+1} with method of high order

$\Rightarrow EST_{k+1} = |\tilde{y}_{k+1}(y_k) - y_{k+1}(y_k)|$.

<u>Rmk</u>: In these two strategies: $y^{\frac{H}{2}}_{k+1}$ or $\tilde{y}_{k+1}$ can be seen as "exact sol."

Since we don't know the exact sol.

(3) Next, we want to improve the strategy ii) above for predicting a good step length for next step by choosing h^*_{k+1}, s.t.

$$EST_{k+1} = TOL_{k+1}$$

If we're given TOL_{k+1}, trivial step length H and method of order p, $p+1$. Then:

One-step error $y_{k+1}(y_k) - y(t_{k+1}, y_k) \approx cH^{p+1}$

and $\tilde{y}_{k+1}(y_k) - y(t_{k+1}, y_k) \approx cH^{p+2}$.

where $y(t; y_k)$ is sol. of IVP with $y(t_k) = y_k$

$\Rightarrow EST_{k+1} = |\tilde{y}_{k+1}(y_k) - y_{k+1}(y_k)| \approx C H^{p+1}$.

So: $C = EST_{k+1} / H^{p+1}$

Set new optimal step $= h^*_{k+1}$. So we require

$C (h^*_{k+1})^{p+1} = TOL_{k+1}$.

$\Rightarrow h^*_{k+1} = H (TOL_{k+1} / EST_{k+1})^{1/p+1}$.

Rmk: To compute h^*_{k+1}, we still need the

eval. of approxi. $\tilde{y}_{k+1}(y_k)$, $y_{k+1}(y_k)$ at $t_k + H$

Algorithm: (No unified/the step length control. algo.)

Let t_0 and y_0 be given. Set $k = 0$
while $(t < T)$ { % Details for step $t_k \to t_{k+1}$

1. For given step length h, compute a local tolerance tol_{k+1}
2. For given step length h, compute $y_{k+1}(y_k)$ and $\tilde{y}_{k+1}(y_k)$, approximations at time t_{k+1} with methods of order p and $p+1$
3. Compute $EST_{k+1} = |\tilde{y}_{k+1}(y_k) - y_{k+1}(y_k)|$
4. If $EST_{k+1} \leq tol_{k+1}$ % Step is accepted
   - $y_{k+1} = \tilde{y}_{k+1}(y_k)$, $t_{k+1} = t_k + h$, $k \to k+1$
   - $h = \max\left(h_{min}, \min\left(\mu h, \varrho h \left(\frac{tol_{k+1}}{EST_{k+1}}\right)^{\frac{1}{p+1}}\right)\right)$
   - if $t_{k+1} + h > T$, then $h = T - t_{k+1}$

   otherwise: % Step is rejected
   - $h = h/2$
   - if $h < h_{min}$ stop the step length control with a warning, otherwise go to 1.

}

Rmk: 1) h_{min} is to avoid the step length

become too small because we prefer

large step for efficiency.

ii) $e \in (0,1)$ is safety factor and $\mu \geq 1$ is amplification factor which multiply on $h^*_{k+1} = h\left(\frac{TOL_{k+1}}{EST_{k+1}}\right)^{1/1+p}$

iii) Error estimate is for method of the order p. But we also need $\tilde{y}_{k+1}(y_k)$

④ <u>Embedded RK method:</u>

We can reduce the computation of strategy above. Since we need to evaluate f twice (in the two methods) $\Rightarrow$ double the cost $\Rightarrow$ expensive

<u>Idea</u>: Reuse the function evaluations.

$$\begin{array}{c|ccc} c_1 & a_{11} & \cdots & a_{1s} \\ \vdots & \vdots & & \vdots \\ c_s & a_{s1} & \cdots & a_{s1} \\ \hline & b_1 & \cdots & b_s \\ \hline & \hat{b}_1 & \cdots & \hat{b}_s \end{array}$$

$\leftarrow$ method of order p.

$\leftarrow$ method of order $p+1$.

<u>e.g.</u> i)

$$\begin{array}{c|cc} 0 & 0 & 0 \\ 1 & 1 & 0 \\ \hline & 1/2 & 1/2 \\ \hline & 1 & 0 \end{array}$$

$\leftarrow$ expl. Trapezoidal

$\leftarrow$ expl. Euler

ii) RK45: combination algorithm of RK4 & RK5 in python.