DC2020/03/30

Monday, March 30, 2020 1:03 PM

4.5.2 Second order RK (RK2) methods

- We formulate EXRK2 (*i.e.*, explicit RK with $s = 2$).

The scheme can be written as

~ quad weights

$$y_{n+1} = y_n + \Delta t\,(b_1 k_1 + b_2 k_2)$$

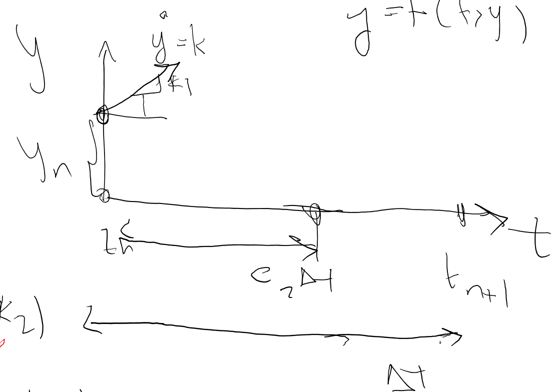
intermediate slopes

$y_{n+1} = y_n + \Delta t$ (average slope)

$\dot{y} = f(t, y)$

explicit 2 stage

$$k_1 = f(t_n, y_n)$$
$$k_2 = f(t_n + c_2\Delta t,\, y_n + \Delta t\, a_{21} k_1)$$



If it was implicit:

$$k_1 = f(t_n + c_1\Delta t,\, y_n + \Delta t\, a_{11} k_1 + \Delta t\, a_{12} k_2)$$
$$k_2 = f(t_n + c_2\Delta t,\, y_n + \Delta t\, a_{21} k_1 + \Delta t\, a_{22} k_2)$$

k_1 & k_2 should be solved through a system of potentially nonlinear eqns

$$y_{n+1} = y_h + \Delta t(b_1 k_1 + b_2 k_2) \qquad \text{where} \tag{264a}$$
$$k_1 = f(t_n, y_n) \tag{264b}$$
$$k_2 = f(t_n + c_2\Delta t, y_n + \Delta t a_{21} k_1) \tag{264c}$$

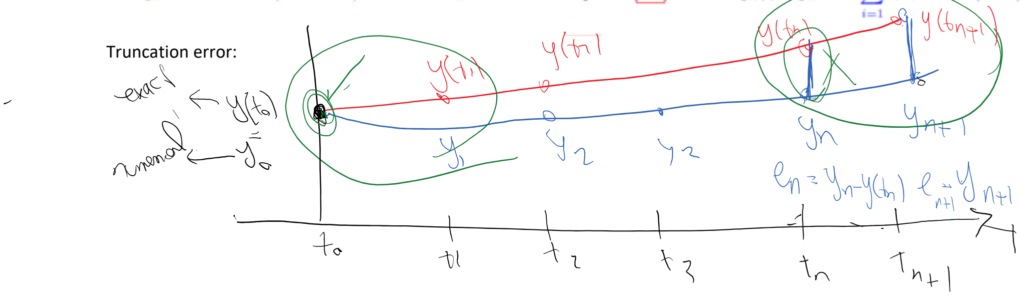
unknowns are b_1, b_2, c_2, a_{21}

- As usual we adopt the following notation,

$y(t_n)$ Exact solution at t_n from the ODE (261): $\frac{dy}{dt} = f(t, y)$ (265a)

y_n RK (numerical) solution at t_n from the RK update (262): $y_{n+1} = y_n + \Delta t \sum_{i=1}^{s} b_i k_i$ (265b)

Truncation error:



$t = T$

$e_n = y_n - y(t_n)$, $e_{n+1} = y_{n+1} - y(t_{n+1})$

$\text{error}(t = T) = \Delta t^q$

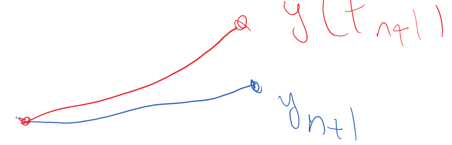
Goal $e(t=T) = ?$

$\text{error}(t = T) = \Delta t^q$ Goal $q = ?$ Goal t @ $(t=1) = r$

final time

To obtain q, a trick (a simpler approach) is to study truncation error:

For truncation error analysis, we assume that the beginning of a step exact and numerical solutions match (as if we are at time step 0)



assumption ← $y(t_n) = y_n$

$$\text{truncation error} = \frac{y_{n+1} - y(t_{n+1})}{\Delta t}$$

y_{n+1} → RK method

$y(t_{n+1})$ → exact solution

Δt → will explain why we divide by Δt later

$y(t_{n+1}) = y(t_n) + \cdots \Delta t$ Taylor's expansion

- The purpose of the analysis in the following is,

Let $y_n = y(t_n)$ (266a)
Update exact solution to t_{n+1} ($y(t_{n+1})$) using (261a). (266b)
Update numerical solution to t_{n+1} (y_{n+1}) using (262). (266c)
Evaluate to what order Δt^q exact and numerical solutions can match by adjusting RK model parameters. (266d)

- First, we evaluate the Taylor expansion of the exact solution from t_n to t_{n+1},

$$y(t_{n+1}) = y(t_n) + \Delta t \frac{\mathrm{d}y}{\mathrm{d}t}(t_n) + \frac{1}{2}\Delta t^2 \frac{\mathrm{d}^2 y}{\mathrm{d}t^2}(t_n) + \cdots + \frac{1}{q!}\Delta t^q \frac{\mathrm{d}^{(q)}y}{\mathrm{d}t^{(q)}}(t_n) + \mathcal{O}\left(\Delta t^{q+1}\right) \quad (267)$$

$\dot{y}(t,y) = f(t,y)$ given ? $\dot y = f$

$$\frac{d^2 y}{dt^2} = \frac{d}{dt}(\dot y) = \frac{d}{dt}\left(f(t,y)\right) = \left(\frac{\partial}{\partial t} f\right)\underbrace{\frac{\partial t}{\partial t}}_{1} + \frac{\partial f}{\partial y}\frac{\partial y}{\partial t}$$

$\boxed{\ddot y = f_{,t} + f_y f}$ Continue this to get $\dddot{y}$, ...

$$\frac{\mathrm{d}y}{\mathrm{d}t}(t_n) := f \quad f \text{ is a shorthand for } f \text{ at } (t_n, y(t_n)) \text{ that is } f = f(t_n, y(t_n)) \text{ (the dependence on } t_n \text{ is not displayed)} \quad (268b)$$

$$\frac{\mathrm{d}^2 y}{\mathrm{d}t^2}(t_n) = \frac{\mathrm{d}f}{\mathrm{d}t}(t_n) = \left(\frac{\partial f}{\partial t}\frac{\mathrm{d}t}{\mathrm{d}t} + \frac{\partial f}{\partial y}\frac{\mathrm{d}y}{\mathrm{d}t}\right)(t_n) \quad (\text{ from (268a)}) \quad \Rightarrow$$

$$\frac{\mathrm{d}^2 y}{\mathrm{d}t^2}(t_n) := f_t + f_y f \qquad \text{note } \frac{\mathrm{d}y}{\mathrm{d}t}(t_n) = f \text{ from (268a) and shorthand notations } f_t := \frac{\partial f}{\partial t}(t_n, y(t_n)), f_y := \frac{\partial f}{\partial y}(t_n, y(t_n)) \quad (268c)$$

$$\frac{\mathrm{d}^3 y}{\mathrm{d}t^3}(t_n) := f_{tt} + f_t f_y + 2 f f_{ty} + f f_y^2 + f^2 f_{yy}, \qquad \text{(obtained in a similar fashion by the use of chain rule)} \quad (268d)$$

- By plugging (268b), (268c), and (268d) in (267) we obtain,

$$y(t_{n+1}) = y(t_n) + \Delta t f + \frac{1}{2}\Delta t^2 \left(f_t + f_y f\right) + \frac{1}{6}\Delta t^3 \left(f_{tt} + f_t f_y + 2 f f_{ty} + f f_y^2 + f^2 f_{yy}\right) + \mathcal{O}\left(\Delta t^4\right) \quad (269)$$

expansion of the exact solution

We need an expansion of the numerical solution in terms of Δt

$y_{n+1} = y_n + \Delta t\,(b_1 k_1 + b_2 k_2)$ $\quad k_1 = f(t_n, y_n) = f$

We need an expansion of the numerical solution in terms of Δt

$k_1 = f(t_n, y_n) = f$

$k_2 = f(t_n + c_2\Delta t, y_n + a_{21}k_1\Delta t)$

$y_{n+1} = y_n + \Delta t(b_1k_1 + b_2k_2)$

$= y_n + \Delta t\,(b_1 \underbrace{f}_{k_1} + b_2 f\underbrace{(t_n + c_2\Delta t, y_n + a_{21}k_1\Delta t)}_{\text{Taylor's expansion of this}}$

$= y_n + \Delta t \left\{ b_1k_1 + b_2\left\{ f(t_n, y_n) + \frac{\partial f}{\partial t}\Big|_{(t_n,y_n)}(c_2\Delta t) + \frac{\partial f}{\partial y}(t_n,y_n)(a_{21}k_1\Delta t) + \frac{1}{2}\frac{\partial^2 f}{\partial t^2}(t_n,y_n)(c_2\Delta t)^2 + \cdots \right\}\right.$

$$
\begin{aligned}
y_{n+1} &= y_h + \Delta t(b_1k_1 + b_2k_2)\\
&= y_h + \Delta t(b_1k_1 + b_2\{f(t_n + c_2\Delta t, y_n + a_{21}\Delta t k_1)\})\\
&= y_h + \Delta t\left(b_1k_1 + b_2\left\{f + [(\Delta t c_2)f_t + (\Delta t a_{21}k_1)f_y] + \left[\frac{1}{2}(\Delta t c_2)^2 f_{tt} + \frac{1}{2}(\Delta t a_{21}k_1)^2 f_{yy} + (\Delta t c_2)(\Delta t a_{21}k_1)f_{ty}\right]\right\}\right)\\
&= y_h + \Delta t\{b_1k_1 + b_2 f\} + \Delta t^2 b_2\{c_2 f_t + a_{21}k_1 f_y\} + \Delta t^3 b_2\left\{\frac{1}{2}c_2^2 f_{tt} + \frac{1}{2}(a_{21}k_1)^2 f_{yy} + c_2 a_{21}k_1 f_{ty}\right\} + \mathcal{O}(\Delta t^4)
\end{aligned}
$$

- Noting that $k_1 = f$ by (264b), we have the final expression for y_{n+1},

$$y_{n+1} = y_h + \Delta t\{b_1 + b_2\} f + \boxed{\Delta t^2 b_2\{c_2 f_t + a_{21} f f_y\}} + \Delta t^3 b_2\left\{\frac{1}{2}c_2^2 f_{tt} + \frac{1}{2}(a_{21}f)^2 f_{yy} + c_2 a_{21} f f_{ty}\right\} + \mathcal{O}(\Delta t^4) \qquad (270)$$

numerical sh

assume they're equal

exact

$$y(t_{n+1}) = y(t_n) + \Delta t f + \boxed{\frac{1}{2}\Delta t^2(f_t + f_y f)} + \frac{1}{6}\Delta t^3(f_{tt} + f_t f_y + 2ff_{ty} + ff_y^2 + f^2 f_{yy}) + \mathcal{O}(\Delta t^4) \qquad (269)$$

$b_1 + b_2 = 1$ $\quad\Delta t$ term

$b_2 c_2 = \frac{1}{2}$

$b_2 a_{21} = \frac{1}{2}$

$\leftarrow \Delta t^2\, b_2(c_2 \boxed{f_t} + a_{21}(ff_y)) = \frac{1}{2}\Delta t^2(\boxed{f_t} + f_y f)$

3 eqns

4 unknowns

- Still we have only three equations in (271c) and four unknowns.
- We let c_2 to be a free parameter and obtain families of EXRK2 methods:

for $c_2 \neq 0$

$$c_2 \neq 0 \qquad (272a)$$

$$b_1 = 1 - \frac{1}{2c_2} \qquad (272b)$$

$$b_2 = \frac{1}{2c_2} \qquad (272c)$$

$$a_{21} = c_2 \qquad (272d)$$

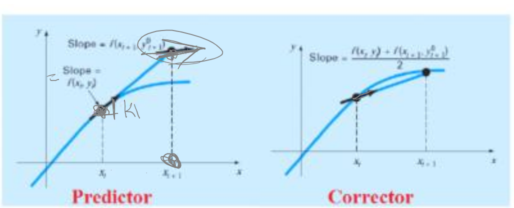
In equation (273) we present some of the well-known members of RK2 methods by assigning different values of c_2.

Name	c_2	RK parameters	RK update	
Heun (Improved Euler)	1	$\begin{bmatrix} b_1 \\ b_2 \\ c_2 \\ a_{21}\end{bmatrix} = \begin{bmatrix} \frac{1}{2} \\ \frac{1}{2} \\ 1 \\ 1\end{bmatrix}$	$\begin{cases} y_{n+1} = y_h + \frac{1}{2}\Delta t(k_1 + k_2) \\ k_1 = f(t_n, y_n) \\ k_2 = f(t_n + \Delta t, y_n + \Delta t k_1)\end{cases}$	(273a)

$y_n + \left(\frac{k_1 + k_2}{2}\right)\Delta t$

forward Euler

$x = t$

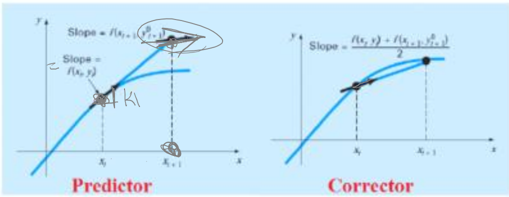


Step 1: slope at the end point is estimated by forward Euler method

Step 2: Slope is updated using predictor equation $y'_{i+1} = f(x_{i+1}, y^0_{i+1})$

Euler

$x = t$



Step 1: slope at the end point is estimated by forward Euler method
$y'_i = f(x_i, y_i)$
End value is estimated
$y^0_{i+1} = y_i + f(x_i, y_i)h$

Step 2: Slope is updated using predictor equation $y'_{i+1} = f(x_{i+1}, y^0_{i+1})$
Take the average slope from 2 y values
$\bar{y}' = \frac{y'_i + y'_{i+1}}{2} = \frac{f(x_i, y_i) + f(x_{i+1}, y^0_{i+1})}{2}$
Update y using the corrector equation

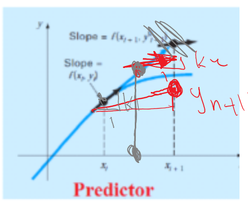
$$y_{i+1} = y_i + \frac{f(x_i, y_i) + f(x_{i+1}, y^0_{i+1})}{2}h$$

Midpoint (Modified Euler) $\frac{1}{2}$

$$\begin{bmatrix} b_1 \\ b_2 \\ c_2 \\ a_{21} \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ \frac{1}{2} \\ \frac{1}{2} \end{bmatrix} \qquad \begin{cases} y_{n+1} &= y_n + \Delta t k_2 \\ \boxed{k_1} &= f(t_n, y_n) \\ k_2 &= f(t_n + \frac{1}{2}\Delta t, y_n + \frac{1}{2}\Delta t k_1) \end{cases} \tag{273b}$$

$y_n + \Delta t(0\, k_1$
$1 k_2)$

go to the midpoint with Forward Euler slope



Does it look like trapezoidal rule (when we wrote the difference equation for mid step)?

t_n $t_{n+\frac{1}{2}}$ t_{n+1}

$\dot{y} = f(t, y)$

$\dot{y}_{n+\frac{1}{2}} = \frac{t_{n+1} - t_n}{\Delta t}$

$y_{n+\frac{1}{2}} = \frac{y_n + y_{n+1}}{2}$

$\dot{y}_{n+\frac{1}{2}} = f\left(t_n + \frac{\Delta t}{2}, \frac{y_n + y_{n+1}}{2}\right)$

Trapezoidal rule

$$\boxed{\frac{y_{n+1} - y_n}{\Delta t} = f\left(t_n + \frac{\Delta t}{2}, \frac{y_n + y_{n+1}}{2}\right)}$$

Implicit scheme (no stability limit) & requires implicit nonlinear solve for y_{n+1}

explicit & has stability limit

$$\begin{cases} y_{n+1} &= y_n + \Delta t k_2 \\ k_1 &= f(t_n, y_n) \\ k_2 &= f(t_n + \frac{1}{2}\Delta t, y_n + \frac{1}{2}\Delta t k_1) \end{cases} \tag{273b}$$

Ralston $\frac{3}{4}$

$$\begin{bmatrix} b_1 \\ b_2 \\ c_2 \\ a_{21} \end{bmatrix} = \begin{bmatrix} \frac{1}{3} \\ \frac{2}{3} \\ \frac{3}{4} \\ \frac{3}{4} \end{bmatrix} \qquad \begin{cases} y_{n+1} &= y_n + \Delta t\left(\frac{1}{3}k_1 + \frac{2}{3}k_2\right) \\ k_1 &= f(t_n, y_n) \\ k_2 &= f(t_n + \frac{3}{4}\Delta t, y_n + \frac{3}{4}\Delta t k_1) \end{cases} \tag{273c}$$

- Ralston [Ralston, 1962, Ralston and Rabinowitz, 1978] determined that choosing $c_2 = \frac{2}{3}$ ($c_2 = \frac{3}{4}$) provides a minimum bound on the truncation error for the second-order RK algorithms.

$$y_{n+1} = y_n + \Delta t\{b_1 + b_2\} f + \Delta t^2 b_2 \{c_2 f_t + a_{21} f f_y\} + \Delta t^3 b_2 \left\{\frac{1}{2}c_2^2 f_{tt} + \frac{1}{2}(a_{21} f)^2 f_{yy} + c_2 a_{21} f f_{ty}\right\} + \mathcal{O}(\Delta t^4) \tag{270}$$

$$y(t_{n+1}) = y(t_n) + \Delta t f + \frac{1}{2}\Delta t^2 (f_t + f_y f) + \frac{1}{6}\Delta t^3 (f_{tt} + f_t f_y + 2 f f_{ty} + f f_y^2 + f^2 f_{yy}) + \mathcal{O}(\Delta t^4) \tag{269}$$

minimize the error between these

terms

- Midpoint (Modified Euler) Uses the y_n to project the solution to the midpoint of the interval, and from there compute the slope k_2 that would project y_n to y_{n+1}. Note that this method is different from trapezoidal rule that is an implicit method and for which the update equation is written for the mid-point of the interval. Mid-point method, is often shown in the shorthand form below,

$$y_{n+1} = y_h + \Delta t\, f\left(t_n + \frac{1}{2}\Delta t, y_n + \frac{1}{2}\Delta t f(t_n, y_n)\right) \qquad \text{Midpoint (Modified Euler)} \tag{274}$$

- Improved Euler's method is a Heun's method without iteration (next figure). The update can be expressed as,

$$y_{n+1} = y_h + \frac{1}{2}\Delta t\left(f(t_n, y_n) + f\left(t_n + \Delta t, y_n + \Delta t f(t_n, y_n)\right)\right) \qquad \text{Heun (Improved Euler)} \tag{275}$$

- To determine the order of accuracy and better understand the behavior of RK2 methods we define the local truncation error $\tau(t_n)$,

exact → numerical

$$\tau(t_n) := \frac{y(t_{n+1}) - y_{n+1}}{\Delta t} \tag{276}$$

leading order term we could n't cancel

$$= \frac{\Delta t^3\left\{\left[\frac{1}{6}\left(f_{tt} + f_t f_y + 2ff_{ty} + ff_y^2 + f^2 f_{yy}\right)\right] - b_2\left[\frac{1}{2}c_2^2 f_{tt} + \frac{1}{2}(a_{21}f)^2 f_{yy} + c_2 a_{21} f f_{ty}\right]\right\} + \mathcal{O}(\Delta t^4)}{\Delta t}$$

$$= \Delta t^2\left\{\left[\frac{1}{6}\left(f_{tt} + f_t f_y + 2ff_{ty} + ff_y^2 + f^2 f_{yy}\right)\right] - \frac{1}{2c_2}\left[\frac{1}{2}c_2^2 f_{tt} + \frac{1}{2}(c_2 f)^2 f_{yy} + c_2 c_2 f f_{ty}\right]\right\} + \mathcal{O}(\Delta t^3)$$

The: order of accuracy of the method is

$\Delta = y_{nf} - y(T)$

$y_{nf} = \mathcal{O}(\Delta t^2)$ for RK2

time steps $= \frac{T}{\Delta t} \times \Delta t^3 = \Delta t^2$

each step error

global error

EXRK2 is as accurate as Trapezoidal method but EXRK2 is explicit and requires linear update equations without solving global systems.

- We observe that the RK2 scheme is second order accurate in time.
- One thing that is clear from (277) is that we could not annihilate the $\mathcal{O}(\Delta t^2)$ term in $\tau(t_n)$ due to the lack of number of parameters for RK2 scheme, even though there was one free unknown value.
- This is often the case with RK schemes, that not parameters of an s-stage RK scheme are used in annihilating factors of Δt^i and for the ones that we can annihilate we often end up with more unknowns that equations. That, is why there may be variants of RK methods for a given stage number s.

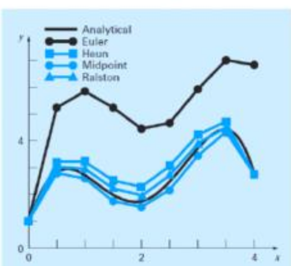


TABLE 25.3 Comparison of true and approximate values of the integral of $y' = -2x^3 + 12x^2 - 20x + 8.5$, with the initial condition that $y = 1$ at $x = 0$. The approximate values were computed using three versions of second-order RK methods with a step size of 0.5.

		Heun		Midpoint		Second-Order Ralston RK	
x	y_{true}	y	$\lvert\varepsilon_t\rvert$ (%)	y	$\lvert\varepsilon_t\rvert$ (%)	y	$\lvert\varepsilon_t\rvert$ (%)
0.0	1.00000	1.00000	0	1.00000	0	1.00000	0
0.5	3.21875	3.43750	6.8	3.109375	3.4	3.277344	1.8
1.0	3.00000	3.37500	12.5	2.81250	6.3	3.101563	3.4
1.5	2.21875	2.68750	21.1	1.984375	10.6	2.347656	5.8
2.0	2.00000	2.50000	25.0	1.75	12.5	2.140625	7.0
2.5	2.71875	3.18750	17.2	2.484375	8.6	2.855469	5.0
3.0	4.00000	4.37500	9.4	3.81250	4.7	4.117188	2.9
3.5	4.71875	4.93750	4.6	4.609375	2.3	4.800781	1.7
4.0	3.00000	3.00000	0	3	0	3.031250	1.0

4.5.3 Fourth order RK (RK4) method

- Perhaps the most popular RK method, is the 4-stage ($s = 4$) fourth order accurate RK4 method below

$$\boxed{y_{n+1} = y_n + \frac{1}{6}\Delta t(k_1 + 2k_2 + 2k_3 + k_4)} \quad \text{where} \tag{278a}$$

$$\begin{aligned} k_1 &= f(t_n, y_n) \\ k_2 &= f(t_n + \tfrac{1}{2}\Delta t, y_n + \tfrac{1}{2}\Delta t k_1) \\ k_3 &= f(t_n + \tfrac{1}{2}\Delta t, y_n + \tfrac{1}{2}\Delta t k_2) \\ k_4 &= f(t_n + \Delta t, y_n + \Delta t k_3) \end{aligned} \tag{278b}$$

t

t_n

$t_{n+1/2}$

t_{n+1}

RK4 is a very popular explicit time marching scheme.

RKDG (RK + discontinuous Galerkin), Cockburn, Shu .... Often RK4 is used

if $\dot{y} = f(t)$ not depend on y

$y_{n+1} = \int_{t_n}^{t_{n+1}} \dot{y}\, dt = \int_{t_n}^{t_{n+1}} f(t)\, dt = y_n + \Delta t \left(\frac{f(t_n)}{6} + \frac{4 f(t_{n+1/2})}{6} + \frac{f(t_{n+1})}{6} \right)$

$k_1 = f(t_n)$

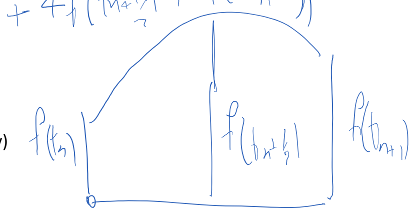
$k_2 = f(t_n + \Delta t/2)$

$k_3 = f(t_n + \Delta t/2)$

$k_4 = f(t_n + \Delta t)$

$= y_n + \frac{\Delta t}{6}\left(f(t_n) + 4 f(t_{n+1/2}) + f(t_{n+1})\right)$

Simpson's rule (from Newton Cote's family)

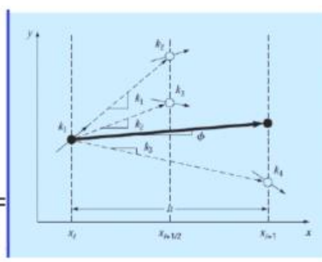


- When derivative is not a function of y, *i.e.*, when $f(t, y) = f(t)$ the solution to the ODE, is simply the integration of a scalar function.

- In such case, RK4 reduces to the Simpson rule for integration of an interval; *cf.* (168):

$$\text{Quadrature}\left(\int_0^L f(x)\,\mathrm{d}x\right) = \frac{L}{6}f(0) + \frac{4L}{6}f(L/2) + \frac{L}{6}f(L)$$

0				
$\frac{1}{2}$	$\frac{1}{2}$			
$\frac{1}{2}$	0	$\frac{1}{2}$		
1	0	0	1	
	$\frac{1}{6}$	$\frac{2}{6}$	$\frac{2}{6}$	$\frac{1}{6}$



$$y_{i+1} = y_i + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)h$$

where

$$\begin{aligned} k_1 &= f(x_i, y_i) \\ k_2 &= f\left(x_i + \tfrac{1}{2}h, y_i + \tfrac{1}{2}k_1 h\right) \\ k_3 &= f\left(x_i + \tfrac{1}{2}h, y_i + \tfrac{1}{2}k_2 h\right) \\ k_4 &= f(x_i + h, y_i + k_3 h) \end{aligned}$$

[Chapra and Canale, 2010]

4.5.4 Butcher effect and higher order RK methods

- From these two results (RK2, RK4) one may be tempted to conclude that the order of accuracy is the same as number of stages s, which is not correct in general.
- The number of unknowns for an s-stage explicit RK method is $s - 1$ (b's) + s (c's) + $(s - 1)s/2$ (a's) = $(s^2 + 3s - 2)/2$.
- The number of equations grow based on what f terms (and in what manner) appear as factors of Δt^i terms. For example, remember that the third order RK expansion was (269).

$$y(t_{n+1}) = y(t_n) + \Delta t f + \frac{1}{2}\Delta t^2 (f_t + f_y f) + \frac{1}{6}\Delta t^3 (f_{tt} + f_t f_y + 2f f_{ty} + f f_y^2 + f^2 f_{yy}) + \mathcal{O}(\Delta t^4)$$

- Unfortunately, there is no guarantee that an s-stage RK method will have s order of accuracy given the different trends the number of equations and unknowns grow and due to the form of the equations.
- For example, if $S(o)$ is the number of RK stages needed for order o we have [Butcher, 1964],

$$N(o) = o \qquad o \le 4 \tag{279a}$$

stages

order of accuracy

$$N(5) = 6 \quad (279b)$$
$$N(6) = 7 \quad (279c)$$

need 6 stages for 5th order accuracy

etc..

- This phenomena is known as the Butcher's effect.
- Given the additional complexity of higher order RK methods and the Butcher's effect (the need of having higher number of stages than order of accuracy) limits the practical uses of higher order RK methods.

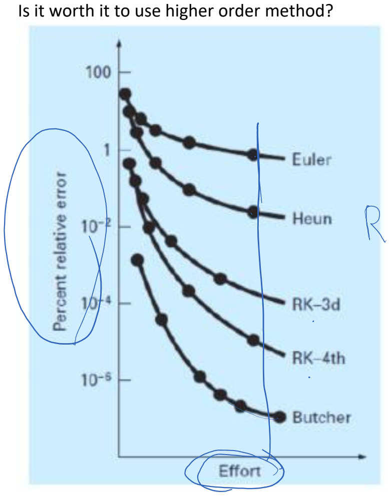
- For the fifth order of accuracy, from (279b) we observe $s = N(o) = N(5) = 6$ stages are required.
- Butcher's fifth order, six-stage RK update equation is given in (280).

$$\boxed{y_{i+1} = y_i + \frac{1}{90}(7k_1 + 32k_3 + 12k_4 + 32k_5 + 7k_6)h}$$

where

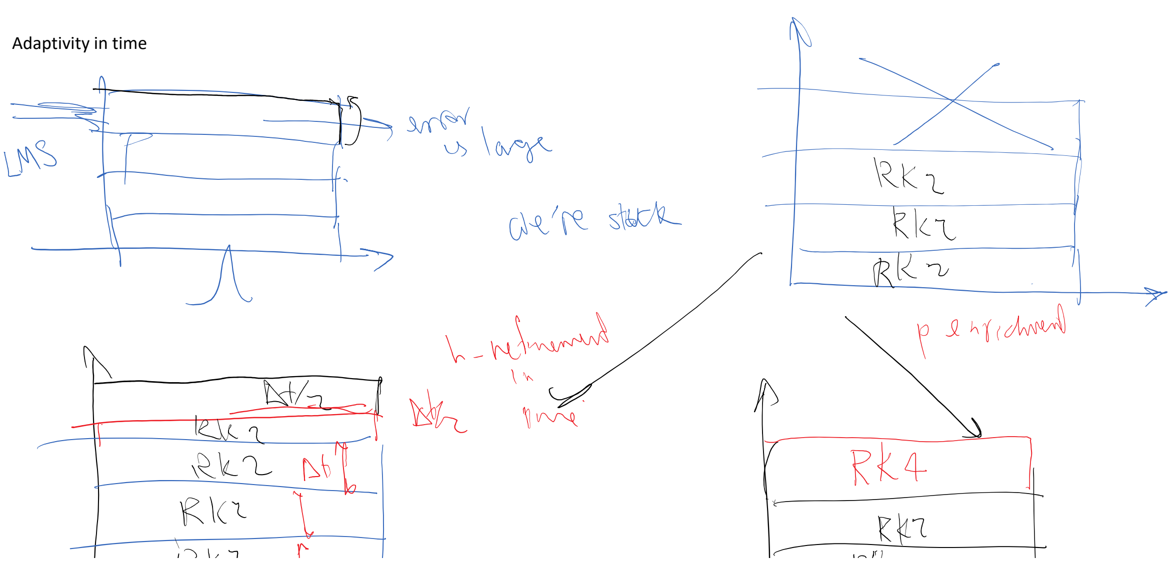
$$k_1 = f(x_i, y_i)$$
$$k_2 = f\left(x_i + \frac{1}{4}h, y_i + \frac{1}{4}k_1h\right)$$
$$k_3 = f\left(x_i + \frac{1}{4}h, y_i + \frac{1}{8}k_1h + \frac{1}{8}k_2h\right)$$
$$k_4 = f\left(x_i + \frac{1}{2}h, y_i - \frac{1}{2}k_2h + k_3h\right)$$
$$k_5 = f\left(x_i + \frac{3}{4}h, y_i + \frac{3}{16}k_1h + \frac{9}{16}k_4h\right)$$
$$k_6 = f\left(x_i + h, y_i - \frac{3}{7}k_1h + \frac{2}{7}k_2h + \frac{12}{7}k_3h - \frac{12}{7}k_4h + \frac{8}{7}k_5h\right) \quad (280)$$

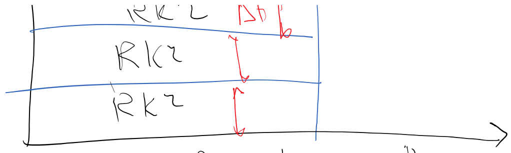
Is it worth it to use higher order method?

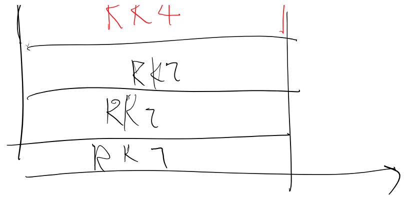


for smooth problem often it's worth going to higher order methods as not only they have higher order conv. rate (and are more accurate) but they are more efficient as well

Adaptivity in time







similar flexibility with variable Time marching
(θ-Wilson Newmark)

A prior error estimate

$$\Delta = \underbrace{y(T)}_{\text{don't have exact s'n}} - y_{nf} = O(\Delta t^{p})$$

A posteriori error estimate: We create something that replaces the exact solution (often in the form of higher order [p-enrichment] or more accurate solution) and from which we calculate a representative error w.r.t. the exact solution (a posteriori error indicator)

- In either case, we need an *a posteriori* error indicator to know
  1. when p-enrichment or h-refinement (when the error is too large) or p-reduction or h-coarsening (h stands for Δt for the time axis) is needed.
  2. which option is more favorable when both p and h options are available. The answer to this question, however is more difficult and in general depends on the regularity of the underlying problem we are solving. Besides for time stepping methods, similar to RK method discussed above, the p-enrichment option is often impractical and we are left with only h-refinement option. Thus, often we do not need to choose between h- or p-adaptivity in time.
- *a posteriori* error indicators: are obtained by the solution of the same time step (or in general local element, update, *etc.*) by comparing the base solution and a more accurate solution. The larger the difference between the two two solutions, the larger the local error.
- Examples for generating more accurate solutions in time, when time stepping methods are used:
  1. Step-halving methods or more generally schemes that cover the same time interval by two different resolutions of time steps. The one with finer step size, clearly represents the more accurate solution scheme.

  2. Different (successive) orders of accuracy: The same time step is solved with two schemes with successive orders of accuracy. The higher order scheme, clearly models the more accurate solution.
- Another use of *a posteriori* error indicators is the ability to improve the accuracy of the solution / or even local order of accuracy by updating the solution with a factor of the a posteriori error. The ability to use the error to improve the accuracy of the solution, requires some mathematical analysis of the time stepping method.
- Below, we present some excepts from [Chapra and Canale, 2010] section 25.5 that discussed both step-halving and different orders of accuracy approaches for formulating an *a posteriori* error indicator.

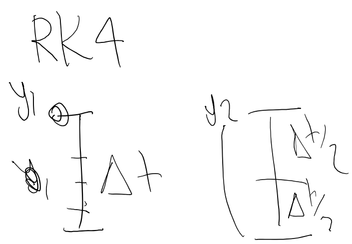
Step halving (also called *adaptive RK*) involves taking each step twice, once as a full step and independently as two half steps. The difference in the two results represents an estimate of the local truncation error. If y_1 designates the single-step prediction and y_2 designates the prediction using the two half steps, the error Δ can be represented as

$$\Delta = y_2 - y_1 \tag{25.43}$$

In addition to providing a criterion for step-size control, Eq. (25.43) can also be used to correct the y_2 prediction. For the fourth-order RK version, the correction is

$$y_2 \leftarrow y_2 + \frac{\Delta}{15} \tag{25.44}$$

This estimate is fifth-order accurate.



more accurate replaces exact s'n