coincide up to order h (independently of the function f). We have (defining $f \equiv f(y_n, x_n)$ $f_x \equiv f_x(y_n, x_n)$ etc)

$$R = (\alpha_1 + \alpha_2) f(y_n, x_n) + \alpha_2 h(\beta_2 f \ f_y + \beta_1 f_x) + \mathcal{O}(h^2)$$

and

$$T = f(y_n, x_n) + \frac{h}{2}(f \ f_y + f_x) + \mathcal{O}(h^2)$$

Equating coefficients at orders 0 and 1 in h gives

$$\begin{aligned} h^0 : \qquad & \alpha_1 + \alpha_2 = 1 \\ h^1 : \qquad & \alpha_2 \beta_2 = \alpha_2 \beta_1 = \frac{1}{2} \end{aligned}$$

and grants that Taylor and RK integration across an elementary discretization interval *independently of the function f and its derivatives* agree up to order h^2. Choosing $\alpha_2 = \gamma$ as free parameter, one obtains $\beta_1 = \beta_2 = \frac{1}{2\gamma}$ and thus the

• **Algorithm**

$$y_{n+1} = y_n + h\left[(1-\gamma) f(y_n, x_n) + \gamma f\left(y_n + \frac{h}{2\gamma} f(y_n, x_n), x_n + \frac{h}{2\gamma}\right)\right] \quad , \qquad x_{n+1} = x_n + h$$

For $\gamma = 1/2$ this algorithm is called Runge-Kutta algorithm of 2nd order (RK2) in the narrow sense. Elementary computational steps are arranged as follows:

• **Runge-Kutta of 2nd order (RK2)**

$$\begin{aligned} k_1 &= h \ f(y_n, x_n) \qquad , \qquad k_2 = h \ f(y_n + k_1, x_n + h) \\ y_{n+1} &= y_n + \frac{1}{2}(k_1 + k_2) \ , \qquad x_{n+1} = x_n + h \end{aligned}$$

In the limit $\gamma \to 0$ (while keeping $h \ll \gamma$) one recovers the Euler-algorithm; the case $\gamma = 1$ defines the so-called Euler-Cauchy algorithm. The following Figure illustrates the precision of the Euler and RK2 algorithms using the simple ODE $y' = y$, with $y(0) = 1$ as initial condition.

RK2 uses 2 function evaluations per discretization interval (instead of 1 for Euler's integration). If a certain precision ϵ at the end of a finite interval's $[a, b]$ is desired, one needs $h = \mathcal{O}(\epsilon)$ for Euler-integration, and needs $\mathcal{O}(\epsilon^{-1})$ function evaluations. For RK2 the requirement is $h^2 = \mathcal{O}(\epsilon)$; with 2 function evaluations per discretization interval this needs $2 \times \mathcal{O}(\epsilon^{-1/2})$ function evaluations (at $\epsilon = 10^{-4}$ this is a gain by a factor 50, at $\epsilon = 10^{-8}$ by a factor 5000!).

Runge-Kutta algorithms of higher order are similarly constructed. The details are somewhat messy, however. Thus we state (without proof) the Runge Kutta Algorithm of 4th order.

• **Runge-Kutta of 4th Order** (RK4)

$$\begin{aligned} k_1 &= h \ f(y_n, x_n) \\ k_2 &= h \ f(y_n + \frac{k_1}{2}, x_n + \frac{h}{2}) \end{aligned}$$