In The Name of Allah

Numerically solving of the SOA

Using Rung-Kutta 4 method

Yaser Khorrami
Y.Khorrami@gmail.com

Runge-Kutta 4th order method is a numerical technique used to solve ordinary differential equation

An ordinary differential equation with Initial value:

$$\frac{dy}{dx} = f(x, y) \qquad y(0) = y_0$$

Goal: find value of unknown function y at a given point x.

The Runge-Kutta method finds approximate value of y for a given x.

Note: Only first order ordinary differential equations can be solved by using the Runge-Kutta 4th order method.

Compute next value y_{n+1} from previous value y_n.

The value of n: 0, 1, 2, 3, …., (x – x0)/h.
h: step height $\mathbf{x_{n+1} = x_0 + h}$

The formula basically computes next value y_{n+1} using current y_n plus weighted average of four increments:

- k_1 is the increment based on the slope at the beginning of the interval, using y
- k_2 is the increment based on the slope at the midpoint of the interval, using y + $hk_1/2$.
- k_3 is again the increment based on the slope at the midpoint, using y + $hk_2/2$.
- k_4 is the increment based on the slope at the end of the interval, using y + hk_3.

The method is a fourth-order method, meaning that the local truncation error is on the order of $O(h^5)$, while the total accumulated error is order $O(h^4)$.

$$y^*(t_0 + h) = y^*(t_0) + \frac{k_1 + 2k_2 + 2k_3 + k_4}{6} h = y^*(t_0) + \left(\frac{1}{6} k_1 + \frac{1}{3} k_2 + \frac{1}{3} k_3 + \frac{1}{6} k_4 \right) h$$

$$= y^*(t_0) + mh \qquad \text{where } m \text{ is a weighted average slope approximation}$$

Example:

$$\frac{dy(t)}{dt} + 2y(t) = 0 \quad \text{or} \quad \frac{dy(t)}{dt} = -2y(t)$$

y(0)=3

$k_1 = -2y^*(t_0)$	approximate derivative at $t = t_0$
$y_1\left(t_0 + \frac{h}{2}\right) = y^*(t_0) + k_1\frac{h}{2}$	intermediate estimate of function at $t = t_0 + h/2$ (using k_1)
$k_2 = -2y_1\left(\frac{h}{2}\right)$	estimate of slope at $t = t_0 + h/2$
$y_2\left(t_0 + \frac{h}{2}\right) = y^*(t_0) + k_2\frac{h}{2}$	another intermediate estimate of function at $t = t_0 + h/2$ (using k_2)
$k_3 = -2y_2\left(\frac{h}{2}\right)$	another estimate of slope at $t = t_0 + h/2$
$y_3(t_0 + h) = y^*(t_0) + k_3 h$	an estimate of function at $t = t_0 + h$ (using k_3)
$k_4 = -2y_3(t_0 + h)$	estimate of slope at $t = t_0 + h$
$y^*(t_0 + h) = y^*(t_0) + \frac{k_1 + 2k_2 + 2k_3 + k_4}{6}h$	estimate of $y(t_0 + h)$

Linear SOA Equations

$$\left\{\begin{aligned} &\frac{\partial N}{\partial t} = \frac{I}{qV} - \frac{N}{\tau_c} - \frac{g(N)}{\hbar\omega_0}|A|^2 \\ &g(N) = \Gamma a(N - N_0) \end{aligned}\right\} \Rightarrow \frac{\partial g}{\partial t} = \frac{g - g_0}{\tau_c} - g\frac{|A|^2}{E_{sat}} \qquad (1)$$

$$E_{sat} = \hbar\omega_0\delta / a \qquad (2)$$

$$\delta = wd / \Gamma \qquad (3)$$

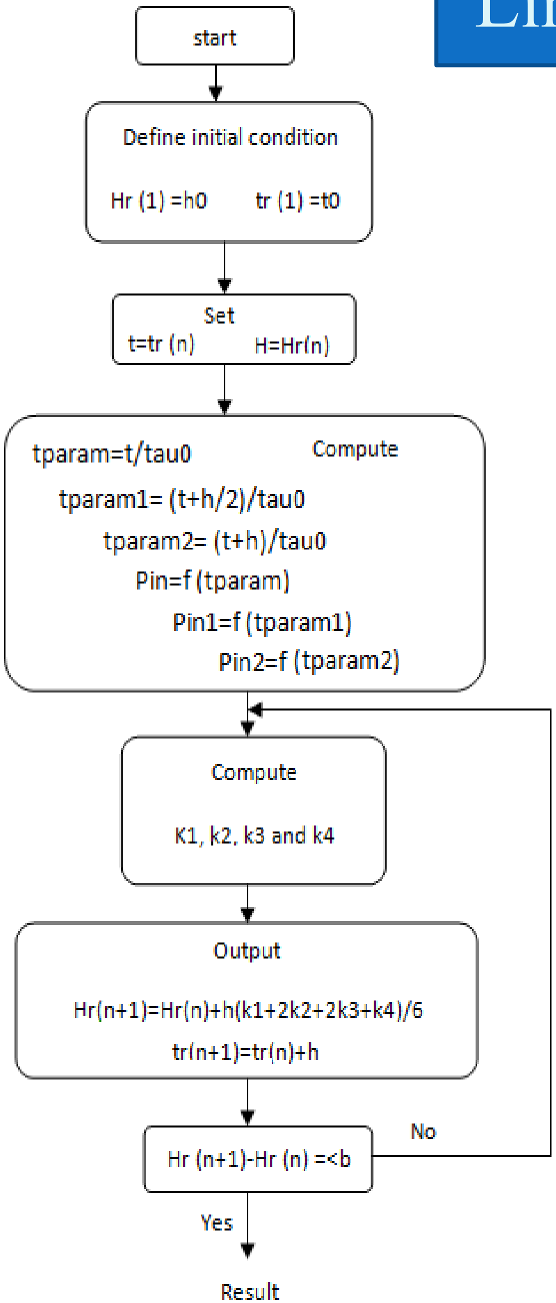
$$I_0 = qVN_0 / \tau_c \qquad (4)$$

$$g_0 = \Gamma a N_0 (I / I_0 - 1) \qquad (5)$$

$$A = \sqrt{P}\exp(i\varphi) \qquad (6)$$

$$\left\{\begin{aligned} &\frac{\partial P}{\partial z} = (g - \alpha_{\text{int}})P && (7) \\ &\frac{\partial \varphi}{\partial z} = -\frac{1}{2}\alpha g && (8) \\ &\frac{\partial g}{\partial t} = \frac{g_0 - g}{\tau_c} - \frac{gP}{E_{sat}} \quad \text{and} \quad h(\tau) = \int_0^L g(z,\tau)dz \Rightarrow \frac{\partial h}{\partial t} = \frac{g_0 L - h}{\tau_c} - \frac{P_{in}(\tau)}{E_{sat}}[\exp(h) - 1] && (9) \end{aligned}\right\}$$

Linear SOA Flowchart by Rung-Kutta 4-order



$$A_{in}(\tau) = \frac{E_{in}}{\tau_0\sqrt{\pi}} \exp\left(-\frac{\tau^2}{\tau_0^{\,2}}\right)$$

$$\tau_p \approx 1.665\tau_0$$

$$A_{in}(\tau) = \frac{E_{in}}{2\tau_0} \sec h^2\left(-\frac{\tau}{\tau_0}\right)$$

$$\tau_p \approx 1.7627\tau_0$$

$$k_1 = \frac{h_0 - h}{\tau_c} - \frac{P_{in}}{E_{sat}}\left[\exp(h) - 1\right]$$

$$k_2 = \frac{h_0 - (H + 0.5hk_1)}{\tau_c} - \frac{P_{in,1}}{E_{sat}}\left[\exp(H + 0.5hk_1) - 1\right]$$

$$k_3 = \frac{h_0 - (H + 0.5hk_2)}{\tau_c} - \frac{P_{in,1}}{E_{sat}}\left[\exp(H + 0.5hk_2) - 1\right]$$

$$k_4 = \frac{h_0 - (H + hk_3)}{\tau_c} - \frac{P_{in,2}}{E_{sat}}\left[\exp(H + hk_3) - 1\right]$$