



KTH Engineering Sciences

Runge–Kutta 4 för system

I Sauer finns iterationsformeln för Runge–Kutta 4

$$\begin{aligned}k_1 &= f(t_n, u_n), \\k_2 &= f\left(t_n + \frac{h}{2}, u_n + \frac{h}{2}k_1\right), \\k_3 &= f\left(t_n + \frac{h}{2}, u_n + \frac{h}{2}k_2\right), \\k_4 &= f(t_n + h, u_n + hk_3), \\u_{n+1} &= u_n + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4),\end{aligned}$$

för att lösa den *skalära* ODEn

$$y' = f(t, y), \quad y \in \mathbb{R}, \quad f \in \mathbb{R}.$$

När man istället vill lösa ett *ODE-system*,

$$\mathbf{y}' = \mathbf{f}(t, \mathbf{y}), \quad \mathbf{y} \in \mathbb{R}^d, \quad \mathbf{f} \in \mathbb{R}^d,$$

använder man exakt samma iterationsformel, men med skalärer utbytta mot vektorer överallt,

$$\begin{aligned}\mathbf{k}_1 &= \mathbf{f}(t_n, \mathbf{u}_n), \\ \mathbf{k}_2 &= \mathbf{f}\left(t_n + \frac{h}{2}, \mathbf{u}_n + \frac{h}{2}\mathbf{k}_1\right), \\ \mathbf{k}_3 &= \mathbf{f}\left(t_n + \frac{h}{2}, \mathbf{u}_n + \frac{h}{2}\mathbf{k}_2\right), \\ \mathbf{k}_4 &= \mathbf{f}(t_n + h, \mathbf{u}_n + h\mathbf{k}_3), \\ \mathbf{u}_{n+1} &= \mathbf{u}_n + \frac{h}{6}(\mathbf{k}_1 + 2\mathbf{k}_2 + 2\mathbf{k}_3 + \mathbf{k}_4).\end{aligned}$$

Notera alltså först att eftersom den exakta lösningen är en vektorvärd funktion $\mathbf{y}(t) \in \mathbb{R}^d$ måste naturligtvis den numeriska approximationen också vara en vektor $\mathbf{u}_n \in \mathbb{R}^d$. Eftersom funktionen $\mathbf{f}(t, \mathbf{y}) \in \mathbb{R}^d$ också är vektorvärd blir även hjälpvariablerna $\mathbf{k}_j \in \mathbb{R}^d$ vektorer.

I MATLAB implementerar ni lämpligen $\mathbf{f}(t, \mathbf{y})$ som en funktion (anonym funktion eller funktionsfil) som returnerar en *vektor*. I tidstegningsloopen kan ni sedan skriva iterationen nästan precis som ovan. Jämför även med koden för Framåt Euler för system som ligger på kurswebben, och hur den skiljer sig från Framåt Euler för skalära ekvationer; samma modifikation behöver göras för Runge–Kutta 4.