Previous Next

9.3.1 2nd order Runge-Kutta Method (RK2)

MO2.4 MO2.7

A popular second-order ($p = 2$) Runge-Kutta (RK2) method is known as the modified Euler method. The basic idea is to attempt to estimate the slope $\mathrm{d}\underline{u}/\mathrm{d}t$ at the middle of a timestep (i.e. at $t^n + \Delta t/2$) and use that estimated slope in the Forward Euler method. Specifically, the modified Euler RK2 method is given by,

$$\underline{a} = \Delta t \underline{f}\left(\underline{v}^n, t^n\right) \tag{9.7}$$

$$\underline{b} = \Delta t \underline{f}\left(\underline{v}^n + \underline{a}/2, t^n + \Delta t/2\right) \tag{9.8}$$

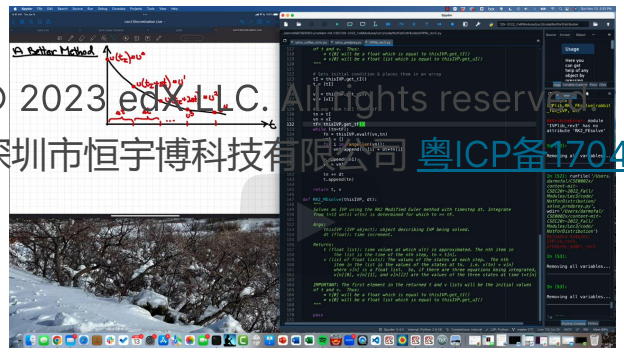
$$\underline{v}^{n+1} = \underline{v}^n + \underline{b} \tag{9.9}$$

The final update involves extrapolating the full timestep from $\underline{v}^n$ using $\underline{b}$. And, $\underline{b}$ is evaluated using the estimated slope at $t^n + \Delta t/2$.

Note that $\underline{f}$ is evaluated twice to calculate the new value of $\underline{v}^{n+1}$: once to determine $\underline{a}$ and once to determine $\underline{b}$. So, per iteration, it is essentially twice as expensive as Forward Euler (since the cost of evaluating $\underline{f}$ usually is the dominant cost of either Forward Euler or modified Euler).

In the following video, we discuss the modified Euler method and implement it in our `IVPlib` code. Then we show the performance of the modified Euler method on the coffee and rabbit-fox problem. The Python codes discussed in this video are available in the following zip file.

Video on modified Euler RK2 method and its implementation



Start of transcript. Skip to the end.

PROFESSOR: We're going to look at a better

Previous Next

Discussions

All posts sorted by recent activity