ph332: starting HW01 + step function review

Nikki Rider
F2021

understanding the code:

what is included?

for all parts

- ode_integrators.py
- utilities.py

functions from these are called in the other files
you do not need to edit them, but do look at them!

for part (a)

- ode_step.py

start here

- lunarlander.py
- kepler.py
- h2formation.py

- you need to modify the get_dydx function to return a numpy array w/ the RHS of the problems diff. eqs.
- the main function is written for you & it calls a function from ode_integrators.py which runs a loop using the step function you choose as a command line option

for part (b)

- error_test.py

for part (c)

- coupled_ode.py

step function review (fixed step integrators)

initial value problem

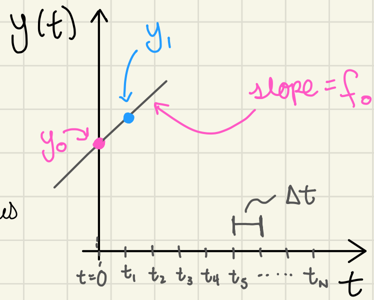
given: $\frac{dy}{dt} = f(t,y)$ ← f RHS in code

$y(0) = y_0$

$\left.\frac{dy}{dt}\right|_{t=0} = f(0,y_0) = f_0$

(will be given actual form of function $f(t,y)$ & constants y_0 & f_0)

• Euler step



$t_j = j\Delta t$

note: sometimes we use "h" instead of "Δt"

* assume the slope (f) is constant between y_0 & y_1 *

$\rightarrow \Delta y = f_0 \Delta t$

the next y-value is the previous plus the change in y

$\Delta y = y_1 - y_0$

setting these equal & solve for y_1 ⟶ first Euler step

$$\boxed{y_1 = y_0 + f_0 \Delta t}$$

general Euler step:

$$\boxed{y_{j+1} = y_j + \Delta t\, f_j}$$

calculated by plugging y_j & t_j into $f(t,y)$

where the notation here is

$y_j \equiv y(t_j)$

$f_j \equiv f(t_j, y_j)$

• RK2 step

from 331:

$$\begin{cases} k_1 = f(t_j, y_j)\,\Delta t \\ k_2 = f(t_j + \alpha\Delta t,\ y_j + \beta k_1)\,\Delta t \\ y_{j+1} = y_j + a k_1 + b k_2 \end{cases}$$

where $a+b=1$, $\alpha b = \frac{1}{2}$, & $\beta b = \frac{1}{2}$

∴ we had some choice in the variables $a, b, \alpha,$ & β.

i like to choose
$a=0$, $b=1$, & $\alpha=\beta=1/2$

$$\boxed{y_{j+1} = y_j + f\left(t_j + \tfrac{1}{2}\Delta t,\ y_j + \tfrac{1}{2}k_1\right)\Delta t}$$

because it has a nice graphical explanation . . .

take a look at k.

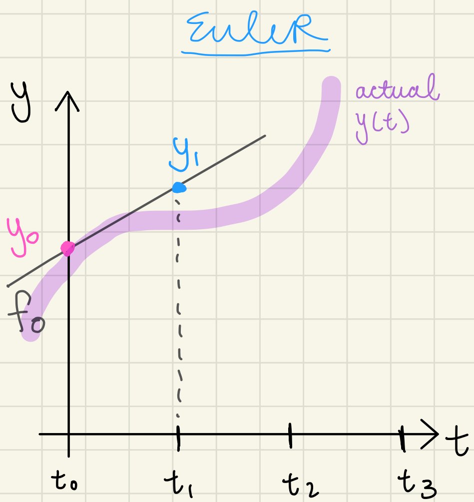
$$k_1 = f(t_j, y_j)\,\Delta t$$

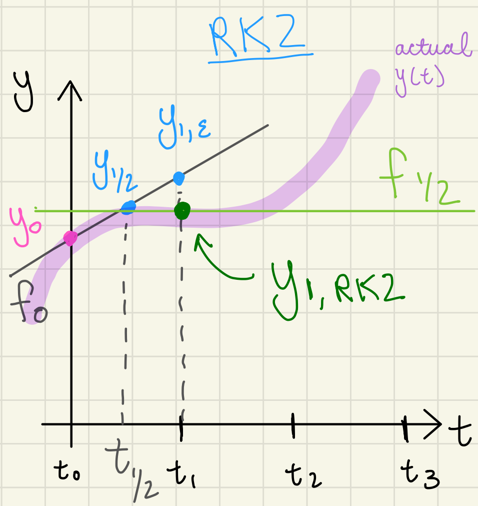
this is just the Euler approx. for Δy between t_j & t_{j+1}

now, looking back at the form of y_{j+1} using RK2, with the constants i chose, for $j = 0$

$$y_1 = y_0 + \underbrace{f\left(\overbrace{t_0 + \tfrac{1}{2}\Delta t}^{t_{1/2}},\ \overbrace{y_0 + \tfrac{1}{2}k_1}^{y_{1/2}}\right)}_{f_{1/2}} \Delta t$$

compare to Euler's first step above. this has the same form as Euler except the slope is evaluated at the midpoint rather than the initial time.





RK2 gives a better estimate of y_1

RK4

i don't have a nice graphical representation for RK4 as of now, but it gives a better estimate than RK2.

$$k_1 = f(t_j, y_j)\Delta t$$

$$k_2 = f\left(t_j + \tfrac{1}{2}\Delta t, y_j + \tfrac{1}{2}k_1\right)\Delta t$$

$$k_3 = f\left(t_j + \tfrac{1}{2}\Delta t, y_j + \tfrac{1}{2}k_2\right)\Delta t$$

$$k_4 = f\left(t_j + \Delta t, y_j + k_3\right)\Delta t$$

$$y_{j+1} = y_j + \frac{1}{6}\left(k_1 + 2k_2 + 2k_3 + k_4\right)$$

---

	local error	global error
Euler	$O(\Delta t^2)$	$O(\Delta t)$
RK2	$O(\Delta t^3)$	$O(\Delta t^2)$
RK4	$O(\Delta t^5)$	$O(\Delta t^4)$