Section 5.4: Runge-Kutta Methods

Thursday, March 18, 2021 8:56 AM

Taylor Methods have nice errors BUT you need to calculate derivatives by hand

IDEA: use (n+1)-point formulas (from §4.1) to approximate derivatives

Taylor's formula for 2-variables

$f(t,y)$?

Want Taylor Polynomial of $f(t,y)$ about the point (t_0, y_0).

Def TP expansion in 2 variable

$$f(t,y) = f(t_0,y_0) + \left[\frac{\partial f}{\partial t}(t_0,y_0)(t-t_0) + \frac{\partial f}{\partial y}(t_0,y_0)(y-y_0)\right]$$
$$+ \left[\frac{\partial^2 f}{\partial t^2}(t_0,y_0)\frac{(t-t_0)^2}{2!} + \frac{\partial^2 f}{\partial y^2}(t_0,y_0)\frac{(y-y_0)^2}{2} + \frac{\partial^2 f}{\partial t \partial y}(t_0,y_0)(t-t_0)(y-y_0)\right] + err$$

GOAL: Start w/ TM2 & replace derivatives.

Recall TM2:

$$T^2(t,y) = f(t,y) + \frac{\Delta t}{2} f'(t,y)$$
$$w_{j+1} = w_j + \Delta t(T^2)$$

Recall TM2 $O(\Delta t^2)$

To not add more error, our approx of f' should be no larger than $O(\Delta t)$

recall by Chain Rule (see §5.3)

$$T^2(t,y) = f(t,y) + \frac{\Delta t}{2}\left[\frac{\partial f}{\partial t} + f(t,y)\frac{\partial f}{\partial y}\right]$$

let's look at Taylor expansion in 2 variables

$$a f(t+b, y+c) = a f(t,y) + a\frac{\partial f}{\partial t}(t,y)[t+b-t]$$
$$+ a\frac{\partial f}{\partial y}(t,y)[y+c-y] + 2^{nd} \text{ deriv error terms}$$
$$= a f(t,y) + ab\frac{\partial f}{\partial t}(t,y) + ac\frac{\partial f}{\partial y}(t,y).$$

let's equalize to T^2 = Taylor Expansion

$$f(t,y) + \frac{\Delta t}{2}\frac{\partial f}{\partial t} + \frac{\Delta t}{2} f(t,y)\frac{\partial f}{\partial y} = af(t,y) + ab\frac{\partial f}{\partial t} + ac\frac{\partial f}{\partial y}$$

$a = 1$ $\quad c = \frac{\Delta t}{2} f(t,y)$

$b = \frac{\Delta t}{2}$

Now we replace RHS of T^2 method with $af(t+b, y+c)$ (which has no derivatives)

RK2
$$\begin{cases} w_0 = \alpha \\ w_{j+1} = w_j + \Delta t\left[f\left(t_j + \frac{\Delta t}{2}, w_j + \frac{\Delta t}{2} f(t_j, w_j)\right)\right] \end{cases}$$

$w_{j+1} = w_j + \Delta t\,[f(t_j, w_j)]$ - Euler's Method

"midpoint method"

$O(\Delta t^2)$

"Runge-Kutta 2"

ex Use RK2 to approximate soln to $y' = y - t^2 + 1$ *Test case*

$0 \le t \le 2$, $y(0) = 0.5$, w/ $N = 10$.

Test case for code

$$\Delta t = \frac{t_f - t_0}{N} = \frac{2-0}{10} = 0.2.$$

$t = 0, 0.2, 0.4, \ldots, 2.$

$w_0 = \alpha,$

$w_0 = 0.5$

$$w_1 = w_0 + \Delta t\left[f\left(t_0 + \frac{\Delta t}{2}, w_0 + \frac{\Delta t}{2} f(t_0, w_0)\right)\right]$$
$$= 0.5 + 0.2\left[f\left(0 + \frac{0.2}{2}, 0.5 + \frac{0.2}{2} f(0, 0.5)\right)\right]$$
$$= 0.5 + 0.2\left[f\left(0.1, 0.5 + 0.1\left[0.5 - 0^2 + 1\right]\right)\right]$$
$$= 0.5 + 0.2[f(0.1, 0.65)]$$
$$= 0.5 + 0.2\left[0.65 - 0.1^2 + 1\right]$$
$$= 0.828$$

← compared to TM2 0.83

The most commonly used RK method is RK4 (Runge-Kutta of order 4) which has accuracy $O(\Delta t^4)$

$= w_j + k_2.$

and is given by:

RK2: w_{j+1}

$$\begin{cases} w_0 = \alpha \\ K_1 = \Delta t f(t_j, w_j) \\ K_2 = \Delta t f(t_j + \frac{\Delta t}{2}, w_j + \frac{1}{2}K_1) \\ K_3 = \Delta t f(t_j + \frac{\Delta t}{2}, w_j + \frac{1}{2}K_2) \\ K_4 = \Delta t f(t_j + \Delta t, w_j + K_3) \\ w_{j+1} = w_j + \frac{1}{6}(K_1 + 2K_2 + 2K_3 + K_4) \end{cases}$$

inside your for loop

$\in$ RK2

| t_j

| midpoint of t_j, t_{j+1}

| t_{j+1}

ex/ Use RK4 to estimate solution to $y' = y - t^2 + 1$ *test case for MATLAB*

$0 \le t \le 2$, $y(0) = 0.5$, w/ $N = 10$.

Test case

$\Delta t = \frac{2-0}{10} = 0.2.$

$j=1$ $w_0 = 0.5$

$K_1 = 0.2 f(0, 0.5) = 0.2[0.5 - 0^2 + 1] = 0.3$

$K_2 = 0.2 f(0 + \frac{0.2}{2}, 0.5 + \frac{1}{2}(0.3)) = 0.2[0.65 - 0.1^2 + 1] = 0.3$

$K_3 = 0.2 f(0 + \frac{0.2}{2}, 0.5 + \frac{1}{2}(0.328)) = 0.2[0.664 - 0.1^2 + 1] = 0.3$

$K_4 = 0.2 f(0.2, 0.5 + 0.3308) = 0.2[0.8308 - 0.2^2 + 1] = 0.358$

$w_1 = 0.5 + \frac{1}{6}[0.3 + 2(0.328) + 2(0.3308) + 0.35816] = 0.82929$

$j=2$

K_1

K_2

K_3

K_4

$w_2 = w_1 + \frac{1}{6}[K_1 + 2K_2 + 2K_3 + K_4]$