Lecture 16

niceguy

October 17, 2022

1 Improving Euler's Method

Recall Euler's method can be thought as approximating an integral

$$\int_{t_n}^{t_{n+1}} f(t,y)dt$$

We can improve Euler's method by finding a better approximation for f on the interval $t \in [t_n, t_{n+1}]$. Our original approximation is that f is approximately $f(t_n, y(t_n))$. A better approximate is to take the average value of f, i.e.

$$f(t,u(t)) \approx \frac{f(t_n, y(t_n)) + f(t_{n+1}, y(t_{n+11}))}{2}$$

However, we do not have access to $y(t_{n+1})$. Instead, we will use the approximated value of $y(t_{n+1})$ using Euler's method, i.e.

$$f(t,y(t)) \approx \frac{f(t_n, y(t_n)) + f(t_{n+1}, y_n + (t_{n+1} - t_n f(t_n, y_n))}{2}$$

We then have

$$y_{n+1} = y_n + \frac{f(_n, y_n) + f(t_{n+1}, y_n + hf(t_n, y_n))}{2} \times h$$

where $h = t_{n+1} - t_n$.
However, there is a downside to this, as more calculations have to be performed for each successive approximation.
The Error comparisons are as follows

Method	Local Truncation Error	Global Truncation Error	Function Evaluations
Euler	h^2	h	1
Improved Euler	h^3	h^2	2

However, we can still improve on this. We first start with the slope $f_1 = f(t_n, y(t_n))$. Extending this to $t_n + \frac{h}{2}$, we get the slope f_2. Now put f_2 at t_n, and extend it to get a second approximation for the slope at $t_n + \frac{h}{2}$, which is f_3. Finally, f_3 is placed at t_n which is then extended to approximate the slope at t_{n_1}, which is f_4. We take a weighted average with f_1, f_2, f_3, f_4 being given weights of $1, 2, 2, 1$ respectively.

$$y_{n+1} = y_n + \frac{s_{n1} + 2s_{n2} + 2s_{n3} + s_{n4}}{6} \times h$$

where

$$\begin{aligned}
s_{n1} &= f(t_n, y_n) \\
s_{n2} &= f(t_n + \frac{h}{2}, y_n + \frac{1}{2}hs_{n1}) \\
s_{n3} &= f(t_n + \frac{h}{2}, y_n + \frac{1}{2}hs_{n2}) \\
s_{n4} &= f(t_n + h, y_n + hs_{n3})
\end{aligned}$$

There are some shortcomings to using a constant step size. There might be certain bounds where a smaller step size (more oscillations) is desired, while there are other bounds where a larger step size is desired (approximately constant). Assuming we want the maximum truncation error to be $\epsilon > 0$. If the error is greater than that, we decrease the step size, and vice versa. We estimate the local truncation error by

$$e_{n+1} \approx |y_{n+1} - z_{n+1}|$$

where z is a better approximation than y.