Errors in Numerical Approximations

Two main types of error

* Global (accumulated) error

this is as a result of

(i) using an approximation formula

(ii) the input y_n at each step is an approximation.

Suppose we use the analytic solution $\phi(t_n)$ at the nth step as input, Then we have only (i) left which gives us the

Local Truncation error (L.T.E.)

* L.T.E is the error that arise as a ~~[illegible]~~ result of using an approximation formula.

Local Truncation Error (L.T.E)

The L.T.E ~~error~~ at the $(n+1)^{th}$ step is given by

$$e_{n+1} = \phi(t_{n+1}) - y_{n+1} \quad \text{—— (1)}$$

$$\phi(t_{n+1}) = \phi(t_n + h)$$

Taylor expand

$$\phi(t_n + h) = \phi(t_n) + h\phi'(t_n) + \frac{h^2}{2}\phi''(t_n) + \cdots \quad (2)$$

From Euler's formula,

$$y_{n+1} = y_n + h f(t_n, y_n) \quad \text{—— (3)}$$

Sub. (2) and (3) into (1).

$$e_{n+1} = \big(\phi(t_n) - y_n\big) + h\big(\phi'(t_n) - f(t_n, y_n)\big)$$

$$+ \frac{h^2}{2}\phi''(t_n) + \cdots$$

But $\phi(t_n) \approx y_n$ and $\phi'(t_n) = f(t_n, y_n)$ for very small h.

$\Rightarrow$

~~that~~ $e_{n+1} \approx \frac{h^2}{2}\phi''(t_n) + \cdots$

$e_{n+1} = h^2 \times (\text{some constant})$ as $h \to 0$.

$\Rightarrow$ For very small h, the L.T.E ~~is proportional~~ scales as h^2.

Suppose we want ~~to~~ solve our IVP on $[0, T]$ ~~with~~ with step-size h.

$\Rightarrow$ ~~so~~ the number steps ~~from~~ from 0 to T is $N = T/h$

and at each step we make an error ~~that~~ is $e_{n+1} \approx h^2 \times \begin{pmatrix}\text{some}\\\text{constant}\end{pmatrix}$

$\therefore$ the global error at ~~the~~ T ~~step~~ is

$$\frac{T}{h} \times h^2 \times \begin{pmatrix}\text{some}\\\text{constant}\end{pmatrix} = h \times \begin{pmatrix}\text{new}\\\text{constant}\end{pmatrix}$$

as $h \to 0$

This implies that the global error scales as h for very small h.

Improved Euler method

For Euler's method, we use the slope at t_n to compute y_{n+1}.

Let us try using the average of the slopes at t_n and t_{n+1}, that is we use

$$\frac{f(t_n, y_n) + f(t_{n+1}, y_{n+1})}{2}. \quad \text{— } (*)$$

Sub. $(*)$ into Euler's formula.

$$y_{n+1} = y_n + h\left[\frac{f(t_n, y_n) + f(t_{n+1}, y_{n+1})}{2}\right] \quad \text{— } (*1)$$

what if $f(t,y)$ is something like

$f = 2ty$, ?

we ~~may~~ have our unknown as input!

What if we compute y_{n+1} using Euler's formula ?

i.e $\boxed{\tilde{y}_{n+1} = y_n + h f(t_n, y_n)}$

Then put in ~~(*1)~~

$$\boxed{y_{n+1} = y_n + \frac{h}{2}\left[f(t_n, y_n) + f(t_{n+1}, \tilde{y}_{n+1})\right]}$$

This method is called <u>the Improved Euler's method</u>

$$\boxed{\begin{gathered} y_{n+1} = y_n + \frac{h}{2}\left[f(t_n, y_n) + f(t_{n+1}, y_n + h f(t_n, y_n))\right] \\ n = 0, 1, 2, \ldots \end{gathered}}$$

We can write this as a two ~~step~~ stage-method that is,

$$\boxed{\begin{aligned} \tilde{y}_{n+1} &= y_n + h f(t_n, y_n) \\ y_{n+1} &= y_n + \frac{h}{2}\left[f(t_n, y_n) + f(t_{n+1}, \tilde{y}_{n+1}) \right] \\ & \qquad n = 0, 1, 2, \ldots \end{aligned}}$$

Doing the same error analysis as we did for Euler's method, we can show that the

L.T.E scales as h^3 as $h \to 0$

and global error scales as h^2

as $h \to 0$.

* This type of ~~multistep~~ multi-stage methods are called <u>Runge–Kutta method (R-K)</u>

* ode 45 in matlab uses 4th order R-K.

Example: use improved Euler's method to approximate the ~~IVP~~ ~~th~~ solution of the IVP

$$y' = 1 - t + 4y, \quad y(0) = 1$$

on $[0,1]$ ~~with~~ with $h = 0.1$

Improved Euler method is given by

$$\tilde{y}_{n+1} = y_n + h f(t_n, y_n)$$

$$y_{n+1} = y_n + \frac{h}{2}\left(f(t_n, y_n) + f(t_{n+1}, \tilde{y}_{n+1})\right)$$

$h = 0.1$, $t_0 = 0$, $y_0 = 1$, $f(t,y) = 1 - t + 4y$.

when $n = 0$,

$$\tilde{y}_1 = y_0 + h f(t_0, y_0) = 1 + 0.1\left(1 - 0 + 4(1)\right)$$
$$= 1.5$$

$$y_1 = 1 + \frac{0.1}{2}\left(5 + \left(1 - 0.1 + 4(1.5)\right)\right)$$

$$= 1.595$$