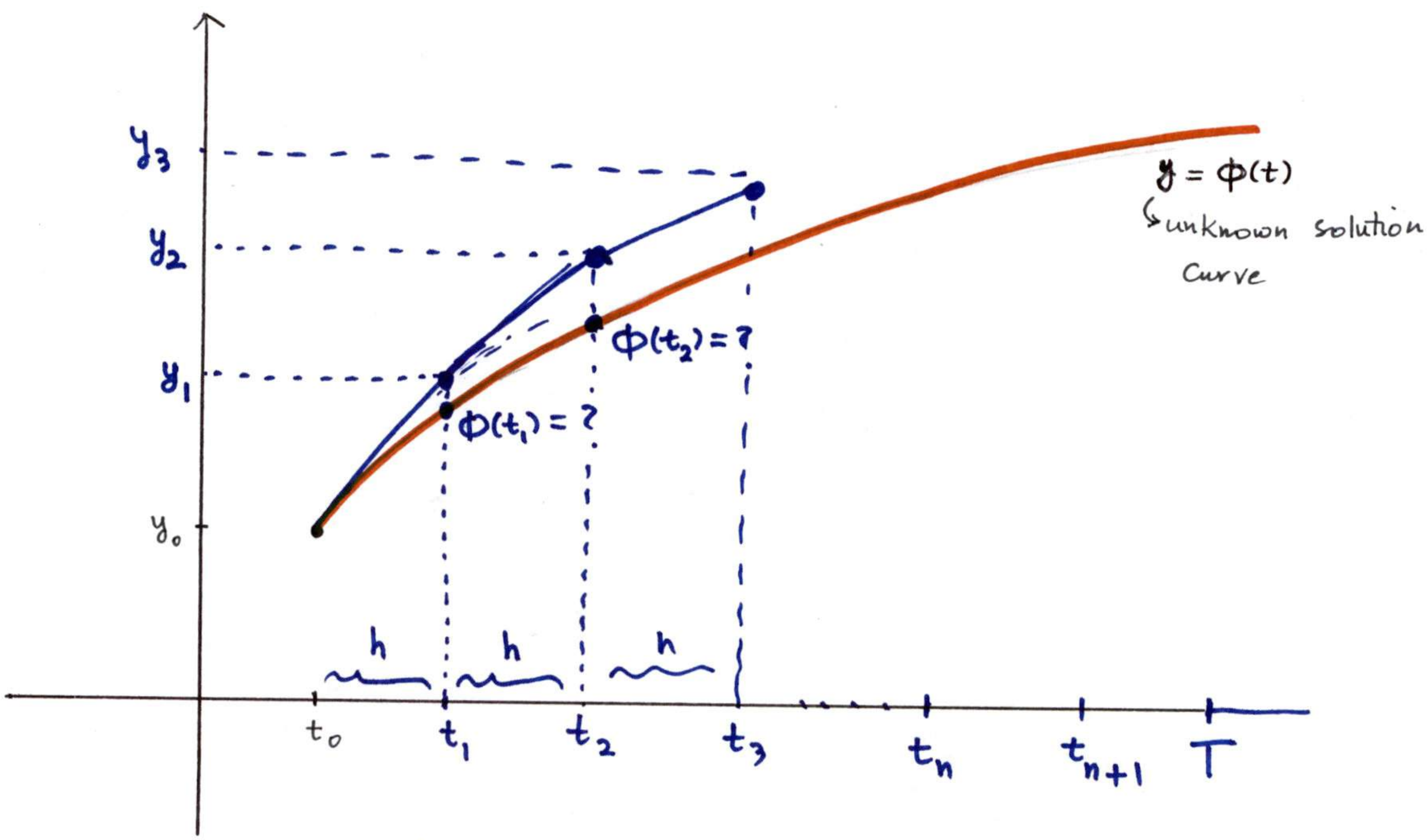
Last Class :

Euler's Method (Numerical Approximation of the Solution)

IVP : $\begin{cases} \dfrac{dy}{dt} = f(t, y(t)) \\ y(t_0) = y_0 \end{cases}$

Let $y = \phi(t)$ be a solution (It is NOT known), by Euler's method we approximate $\phi(t)$ at different values of t.



h = step size , $t_1 = t_0 + h$, $t_2 = t_0 + 2h$, ..., $t_n = t_0 + nh$

# of steps = # of subintervals : $N = \dfrac{T - t_0}{h}$

Procedure

→ Start at (t_0, y_0) and find the tangent line at t_0 :

By IVP $\quad m_{tan} = \left.\dfrac{dy}{dt}\right|_{(t_0, y_0)} = f(t_0, y_0)$

$$\Rightarrow \quad y = y_0 + f(t_0, y_0)(t - t_0)$$

Continue for a short interval (t_0, t_1) ⇝ $y_1 = y_0 + f(t_0, y_0)\overbrace{(t_1 - t_0)}^{h}$
and evaluate the tangent line at t_1

Actual Tangent line at t_1 will be: $\quad m_{tan} = \left.\dfrac{dy}{dt}\right|_{(t_1, \phi(t_1))} = f(t_1, \phi(t_1))$

$$y = \phi(t_1) + f(t_1, \phi(t_1))(t - t_1)$$ ⇝ This is tangent to $\phi(t)$ at $(t_1, \phi(t_1))$

$\phi(t_1)$ is the actual value which is NOT known so take

$$\phi(t_1) \approx y_1$$

then

$y = y_1 + f(t_1, y_1)(t - t_1)$ ⇝ This is the tangent line to some nearby curve at (t_1, y_1)

evaluate at t_1

$y_2 = y_1 + f(t_1, y_1)\overbrace{(t_2 - t_1)}^{h}$

$y_3 = y_2 + f(t_2, y_2)\,h$

⋮

succession of tangent line approximations and each y_n lies on the approximated tangent line.

$$\boxed{y_{n+1} = y_n + f(t_n, y_n)\,h}$$

→ Euler's formula
h: step size on (t_0, T)

N: # of steps ⇒ $\left| N = \dfrac{T - t_0}{h} \right|$

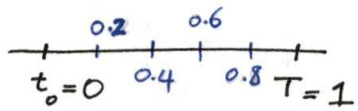
Example: Consider IVP $\begin{cases} \dfrac{dy}{dt} = \underbrace{\boxed{3 - 2t - 0.5y}}_{f(t,y)} \\ y(\underset{t_0}{0}) = 1 \end{cases}$

Use Euler's method to estimate $y(\underset{T}{1})$ with 5 steps.

5 steps : $N = 5$

interval $(0,1)$

$\Rightarrow$ step size $h = \frac{1}{5} = 0.2$



$\boxed{n=0} \rightsquigarrow (0, 1) = (t_0, y_0)$

$$y_1 = y_0 + f(t_0, y_0) h$$

$$\Rightarrow y_1 = 1 + (3 - 0.5)\, 0.2 = 1.5$$

$\boxed{n=1} \rightsquigarrow (0.2, 1.5) = (t_1, y_1)$

$$y_2 = y_1 + f(t_1, y_1) h$$

$$= 1.5 + \big(3 - 2(0.2) - 0.5(1.5)\big)\, 0.2 = 1.87$$

$$\Rightarrow y_2 = 1.87$$

$\boxed{n=2} \rightsquigarrow (0.4, 1.87)$

$$y_3 = 1.87 + \big(3 - 2(0.4) - 0.5(1.87)\big) \cdot 0.2$$

$\boxed{n=3} \rightsquigarrow (0.6, y_3)$

$$y_4 = \ldots$$

$\boxed{n=4} \rightsquigarrow (0.8, y_4)$

$$y_5 = \ldots$$

$\boxed{n=5} \rightsquigarrow (1, y_5)$

$$\boxed{y_6 = y(1) = 2.32363}$$

⊛ Solve IVP analytically
↳ linear 1st order

$$y = \phi(t) = 14 - 4t - 13e^{-\frac{t}{2}}$$

Exact value $\phi(1) = 2.1151$

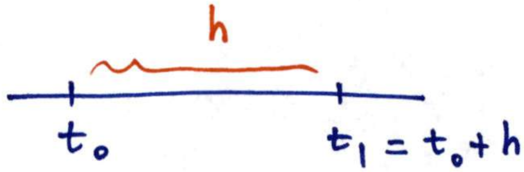
Error of approximation :

$$|\phi(1) - y(1)| = 0.20853$$

quite big error

Error Analysis in Euler's method :

Error in one step going from t_0 to t_1



Actual value at t_1 : $\phi(t_1)$

Approximate value at t_1 : $y_1 = y_0 + f(t_0, y_0)h$

Assume ϕ is a nice function (twice differentiable) so that it has a valid Taylor expansion around t_0

$$\phi(t) = \phi(t_0) + \phi'(t_0)(t - t_0) + \frac{\phi''(t_0)}{2}(t - t_0)^2 + \cdots$$

take $t = t_1 = t_0 + h$

$$\phi(t_1) = \phi(t_0) + \phi'(t_0)h + \frac{\phi''(t_0)}{2}(h^2) + \cdots$$

$$\phi(t_1) = \underbrace{y_0 + f(t_0, y_0)h}_{y_1} + \frac{\phi''(t_0)}{2}h^2 + \cdots$$

$\boxed{\dfrac{dy}{dt} = f(t, y(t)) \quad \hookrightarrow \phi'(t_0) = f(t_0, y_0)}$

$$\underbrace{\phi(t_1) - y_1}_{\text{error in one step}} = h^2 \text{ (some stuff)}$$

<u>Conclusion</u>: $\left\{ \begin{array}{l} \text{error in} \\ \text{one step} \end{array} \right.$ $e_n \propto h^2$

local truncation error

* local error is proportional to h^2.

This means

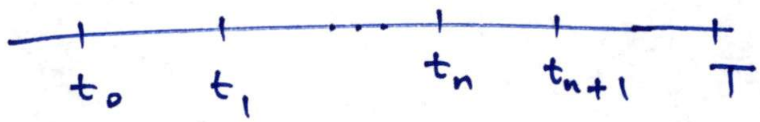
start with $h \rightsquigarrow e_n$

change h to $\frac{h}{2} \rightsquigarrow e_n$ reduces $\frac{1}{4}$ of its original value

$$e_n \downarrow \frac{1}{4} e_n$$

<u>Global truncation error</u> :

go from t_0 to T



N : # of steps $\rightsquigarrow N = \frac{T - t_0}{h}$

$$\underset{\text{error}}{\text{Global}} \; E_n = \underset{\text{in 1 step}}{\text{local error}} \times \text{ \# of steps}$$

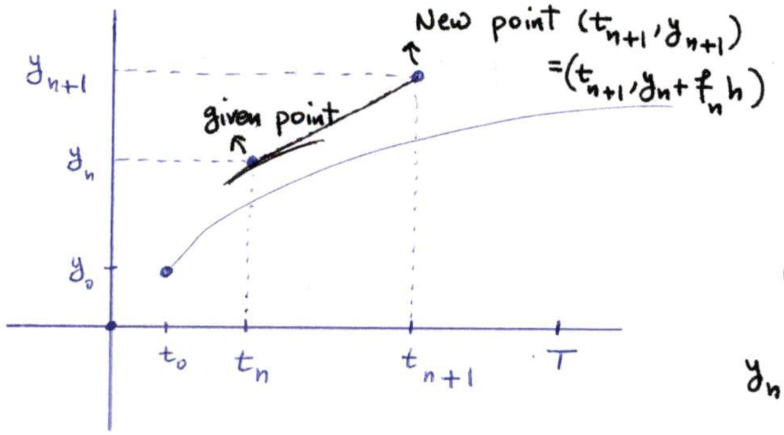
$$= h^2 \times (\text{Some stuff}) \times \frac{T - t_0}{h}$$

$$= h \times (\text{some other stuff})$$

$\boxed{E_n \propto h} \to$ Euler's method is a $\boxed{1^{st} \text{ order method}}$

b/c global error is proportional to the $\underline{1^{st}}$ <u>power of the step size</u>

Improved Euler Method



* In Euler: at each step (t_n, y_n) is known, we find the point (t_{n+1}, y_{n+1}) to be a point on the tangent line to some imaginary nearby curve at (t_n, y_n):

$$y_{n+1} = y_n + f(t_n, y_n) h$$
$$= y_n + f_n h$$

But in Improved Euler:

* Start with (t_n, y_n) $\xrightarrow{\text{Apply Euler}}$ $y_{n+1} = y_n + f(t_n, y_n) h$

$\downarrow$ find slope

$$m_1 = f(t_n, y_n)$$

* But don't ake it the new point

$\downarrow$ find slope

$$m_2 = f(t_{n+1}, y_{n+1})$$
$$= f(t_{n+1}, y_n + f(t_n, y_n) h)$$

Find average of the two slopes

$$m = \frac{1}{2}(m_1 + m_2)$$

Take the new point to be:

$$y_{n+1} = y_n + \frac{1}{2}(m_1 + m_2) h$$

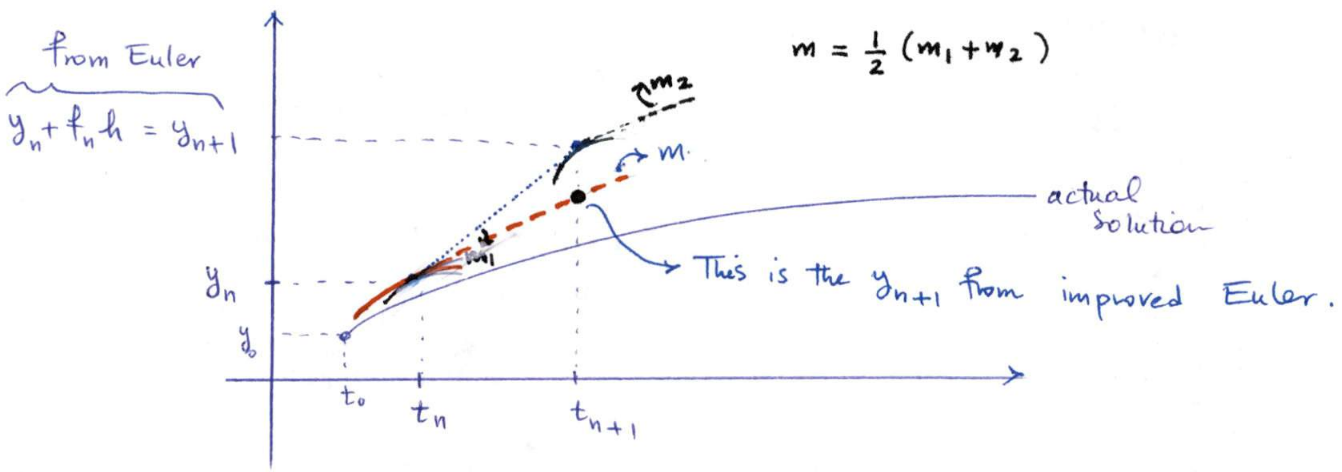
$\Rightarrow$

$$\boxed{y_{n+1} = y_n + \frac{h}{2}\Big(f(t_n, y_n) + f\big(t_{n+1}, y_n + f(t_n, y_n) h\big)\Big)}$$

Improved Euler formula.

__Geometrically__ consider $m_1 = f(t_n, y_n)$ to be the slope of tangent line to some nearby (near actual solution) curve at (t_n, y_n) and consider $m_2 = f(t_{n+1}, y_{n+1})$ to be the slope of tangent line to some nearby curve at (t_{n+1}, y_{n+1}) where y_{n+1} has been obtained from Euler's method.

Now construct another tangent line at (t_n, y_n) but this time with slope $m = \frac{1}{2}(m_1 + m_2)$, then the approximate value y_{n+1} (different from y_{n+1} above) will be the point on this new constructed tangent line.



* It can be shown that Improved Euler is in fact an improvement to Euler since

$$\text{local error } e_n \propto h^3$$

$$\& \text{ global error } E_n \propto h^2 \rightsquigarrow \text{Improved Euler is a } \underline{2^{nd} \text{ order}} \text{ method.}$$

This is different from Order of ODE.

* Note that this reduction of the error is at the cost of more computation Companing to Euler.

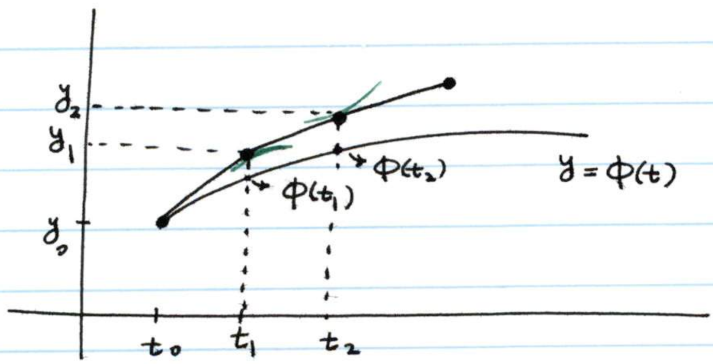
In Improved Euler, we evaluate the function $f(t,y)$ twice, going from t_n to t_{n+1}.

* The most frequent commonly used in real applications is "4th order Runge-Kutta" method, in which

$$e_n \propto h^5 \quad \text{and} \quad E_n \propto h^4 .$$

  * This method in MATLAB is usually done by ODE45( ) which implements a variation of Runge-Kutta (variable time steps to minimize the extra computations of modifying the time step size.

Notes on Euler :



Actual tangent line at $(t_1, \phi(t_1))$:

$$y = \phi(t_1) + f(t_1, \phi(t_1))(t - t_1)$$

but $\phi(t_1)$?

so $\phi(t_1) \approx y_1$

$$\Rightarrow y = y_1 + f(t_1, y_1)(t - t_1)$$

$$\Rightarrow y_2 = y_1 + f(t_1, y_1)\underbrace{(t_2 - t_1)}_{h}$$

At the 2^{nd} step, the tangent line is not tangent to $y = \phi(t)$ but to some near by curve say $y = \phi_1(t)$ that passes through (t_1, y_1)

$\Rightarrow$ Euler's method uses a succession of tangent line approximations to a sequence of different solutions $\phi(t), \phi_1(t), \phi_2(t), \cdots$ of the IVP (with different initial values).

$$\begin{cases} \dfrac{d\phi_1}{dt} = f(t, \phi_1(t)) \\ \phi(t_1) = y_1 \end{cases} \quad \cdots \quad \begin{cases} \dfrac{d\phi_n}{dt} = f(t, \phi_n(t)) \\ \phi(t_n) = y_n \end{cases}$$

* At each step, imagine nearby curves (solutions) starting at (t_n, y_n) and take the tangent line to that curves at (t_n, y_n) and y_{n+1} is the point on this tangent line corresponding to t_{n+1}.

* We get a sequence of points $y_0, y_1, y_2, \cdots, y_{n+1}, \cdots$ that approximate $\phi(t_0), \phi(t_1), \cdots, \phi(t_{n+1})$.

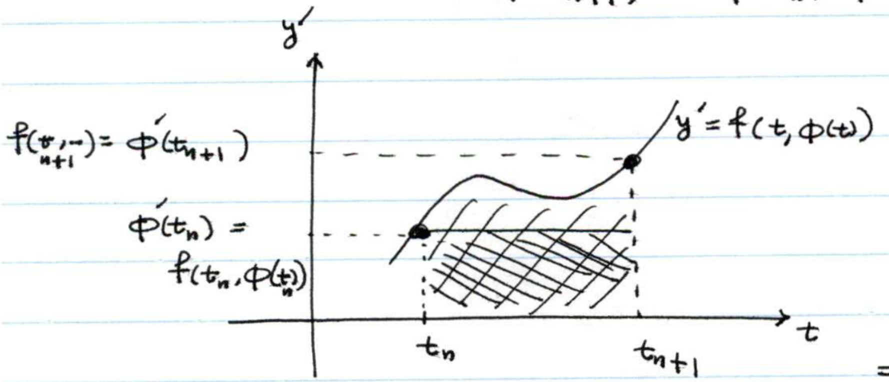
* We get a sequence of line segments (each tangent to a nearby curve around the solution $\phi(t)$) and the piecewise function constructed by these lines is an approximate function for the solution $\phi(t)$.

Another way to look at Euler

$y = \phi(t)$ is a solution: $\dfrac{d\phi}{dt} = f(t, \phi(t))$

integrate over (t_n, t_{n+1}):

$$\phi(t_{n+1}) = \phi(t_n) + \underbrace{\int_{t_n}^{t_{n+1}} f(t, \phi(t))\, dt}_{\text{Area under the curve below}}$$



Replace $\phi(t)$ by $\phi(t_1)$

and $f(t, \phi(t)) \approx f(t_n, \phi(t_n))$

$\Rightarrow$ Area of the rectangle

$$\Rightarrow \quad \phi(t_{n+1}) \approx \phi(t_n) + f(t_n, \phi(t_n))\overbrace{(t_{n+1} - t_n)}^{h}$$

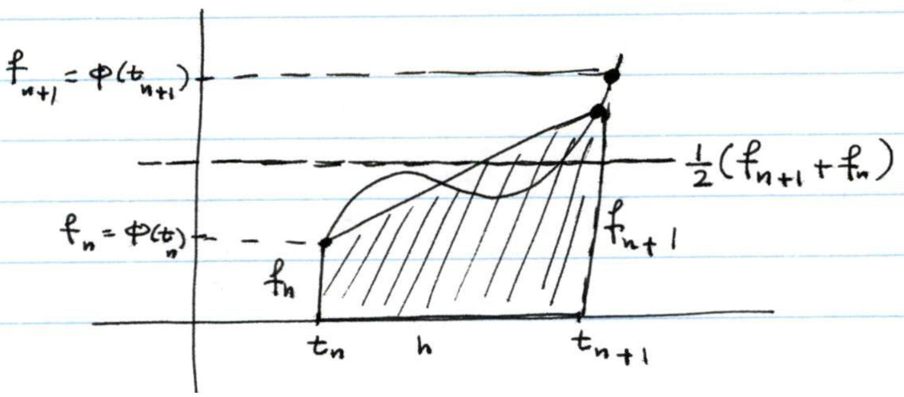
$$\xRightarrow{\phi(t_n) \approx y_n} \quad y_{n+1} \approx \phi(t_{n+1}) \approx y_n + f(t_n, y_n)\, h$$

$$\phi(t_{n+1}) \approx y_{n+1} = y_n + f(t_n, y_n)\, h$$

<u>Improved Euler</u> is a more accurate method of approximating the area under the curve:

In Euler: $\displaystyle\int_{t_n}^{t_{n+1}} \underbrace{f(t, \phi(t))}_{\approx f(t_n, \phi(t_n))}\, dt = f(t_n, \phi(t_n))\, h \approx f(t_n, y_n)\, h$

Improved Euler: $f(t, \phi(t)) \approx \frac{1}{2}\Big(f(t_n, \phi(t_n)) + f(t_{n+1}, \phi(t_{n+1}))\Big)$



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

$$\approx \frac{1}{2}(f_n + f_{n+1}) \cdot h$$

Area of the trapzoid

then $$y_{n+1} = y_n + \frac{h}{2}\Big(f(t_n, \underbrace{\phi(t_n)}_{\approx y_n}) + f(t_{n+1}, \underbrace{\phi(t_{n+1})}_{\approx y_{n+1}})\Big)$$

$$\Rightarrow \quad y_{n+1} = y_n + \frac{h}{2}\Big(f(t_n, y_n) + f(t_{n+1}, y_{n+1})\Big)$$

↳ Implicit formula because y_{n+1} appears on RHS
might be fairly diffcult to solve for y_{n+1}

↳ Use Euler and replace y_{n+1} on RHS by Euler:

$$\boxed{y_{n+1} = y_n + \frac{h}{2}\Big(f(t_n, y_n) + f(t_n, y_n + h f(t_n, y_n))\Big)}$$

* if $f(t,y) = f(t)$

$$\phi_{n+1} \approx y_{n+1} = y_n + h \int_{t_n}^{t_{n+1}} f(t)\,dt \rightarrow \frac{1}{2}\big(f(t_n) + f(t_{n+1})\big)$$

$$= y_n + \frac{h}{2}\big(f(t_n) + f(t_{n+1})\big)$$

↳ Trapezoid rule for numerical integration.