☆ Numerical Approximation:

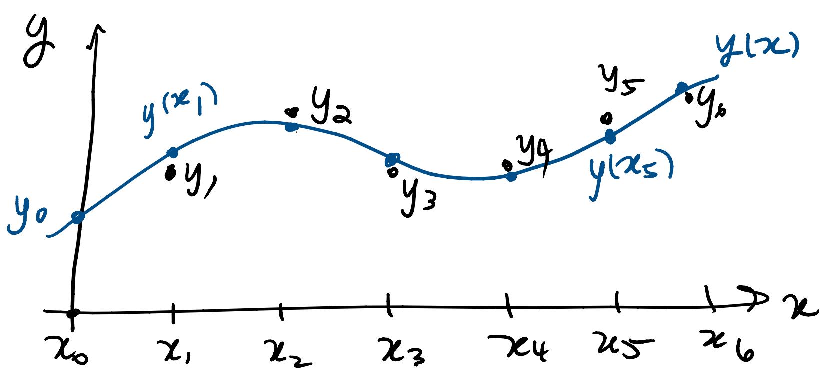
Warm up: Write down the definition of the derivative of the function $f(x)$

$$\frac{df}{dx} = \lim_{h \to 0} \underbrace{\frac{f(x+h) - f(x)}{h}}_{\text{finite difference}}$$

I. Numerical Approximation:

GOAL: Approximate a solution numerically

$$\frac{dy}{dx} = f(x,y), \qquad y(0) = y_0$$



Assume $y(x)$ solves the IVP

$\{y_n\}_{n=0}^{6}$, $\{x_n\}_{n=0}^{6}$ — discrete function

GOAL: Find a formula for y_n so that y_n is close to $y(x_n)$

Choose $\{x_n\}_{n=1}^{6} \longrightarrow$ equally spaced

$$x_n = n \cdot h$$

$$x_0 = 0,\quad x_1 = h,\quad x_2 = 2h,\quad \vdots \quad x_n = n \cdot h$$

h is called the <u>step size</u>

Euler's Method:

approximate the ODE: $\frac{dy}{dx} = f(x,y)$, $y(0) = y_0$

Recall: $$\frac{dy}{dx} = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h}$$

Approximate the derivative by finite difference

$\frac{dx}{} \quad h\to 0 \quad h$

Approximate the derivative by finite difference

$$f(0,y(0)) = \left.\frac{dy}{dx}\right|_{x=0} \approx \frac{y(h)-y(0)}{h} \approx \frac{y_1-y_0}{h}$$

$$\frac{y_1-y_0}{h} = f(x_0,y_0)$$ Euler formula

Do this for each x_n

$$\left.\frac{dy}{dx}\right|_{x=x_n} \approx \frac{y_{n+1}-y_n}{h} = f(x_n,y_n)$$

Rearrange gives a formula

$$\begin{cases} y_{n+1} = y_n + h\,f(x_n,y_n) \\ y_0 = y_0 \end{cases}$$ ← initial condition

<u>Recursion Relation</u>

use y_0 to get y_1

use y_1 to get y_2 .... and so on.

<u>Ex</u>: $\frac{dy}{dx} = \underline{-y}$ $\qquad y(0)=1$

(We know the solution $y(x) = e^{-x}$)

Calculate approximation w/ step size $h=0.1$

$y_0 = 1$ $\qquad$ Here $f(x,y) = -y$

$$y_1 = y_0 + h\,f(x_0,y_0)$$
$$= y_0 + h(-y_0) = \underline{(1-h)y_0}$$
$$= (1-0.1)(1) = (0.9)(1) = 0.9 \qquad \boxed{y_1 = 0.9}$$

$$y_2 = y_1 + h\,f(x_1,y_1) = y_1 + h(-y_1)$$
$$= \underline{(1-h)y_1} = (1-0.1)(0.9) = (0.9)(0.9)$$
$$\boxed{y_2 = 0.81}$$

$$\boxed{y_2 = 0.81}$$

$\vdots$

See a pattern

$$y_n = (1-h)\,y_{n-1} = (1-h)^{n-1}\,y_0 = (0.9)^n$$

$$\boxed{y_n = (0.9)^n}$$

We know that $\lim_{x\to\infty} y(x) = \lim_{x\to\infty} e^{-x} = 0$ ✓

But also $x_n \to \infty$, $h \to \infty$ $\quad \lim_{n\to\infty} y_n = \lim_{n\to\infty} (0.9)^n = 0$

$(x_n = nh)$

★ Code Euler's Method:

use MATLAB, any programing language

$\frac{dy}{dx} = -y$, $y(0) = 1$

$y(1) \approx ?$

PSEUDOCODE:

```
// Initialize variables
y0 = 1,  x0 = 0     xend = 1,     h = 0.1
(*)   N = ((xend - x0)/h) + 1              // N - number of steps.

      xn = array of length N
      yn = array of length N

(*)   xn(0) = x0
      yn(0) = y0

// Euler's method
for j=1 to N
    xn(j) = j*h
(□) yn(j) = yn(j-1) + h * f(xn(j-1), yn(j-1))
end
```

careful with indices (MATLAB - array starts at 1)

// print out $y(1) \approx$
print yn(N)

Two options for calculating f

(a) Write a function $f(x,y)$ that returns the value of y'

```
function yp = f(x,y)
    yp = -y
end
```

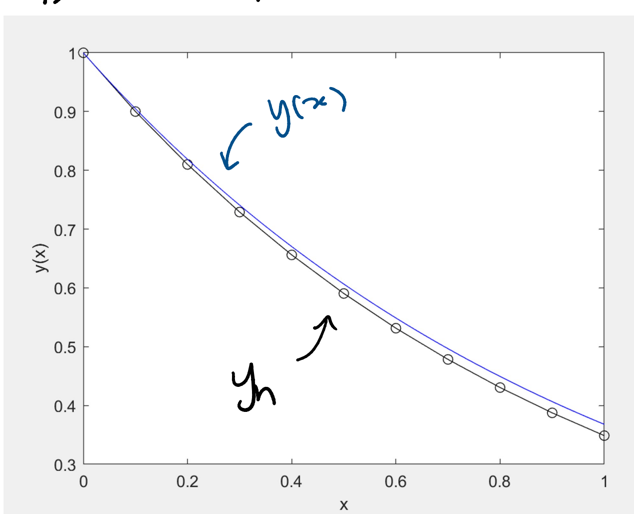
(b) Hard code the definition of $f(x,y) = -y$

(B) $yn(j) = yn(j-1) + h * \underbrace{(-yn(j-1))}$

Either way works.

→ convert code into programming language of your choice.

Matlab code Plot

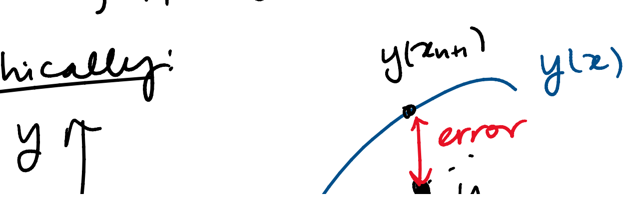


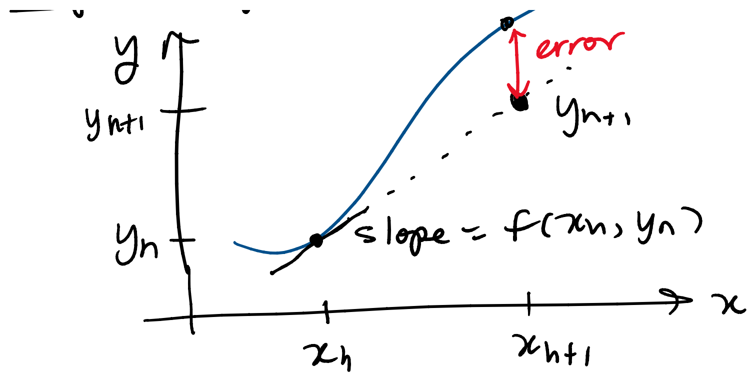
code on B.S.

Q: How accurate is Euler's method?

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

Graphically:





$$\frac{dy}{dx} = f(x,y)$$

- use slope to predict next point

Euler's method is not "super" accurate especially when h is large

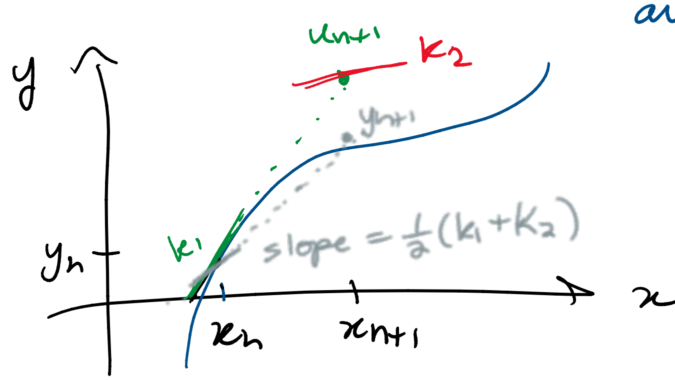
II. Improved Euler's Method:

$$\frac{dy}{dx} = f(x,y) \qquad y(0) = y_0$$

Idea: use 2 estimates of the slope to predict y_{n+1}

Formula:

$$\begin{cases} k_1 = f(x_n, y_n) & \leftarrow \text{1st slope estimate} \\ u_{n+1} = y_n + h k_1 & \\ k_2 = f(x_{n+1}, u_{n+1}) & \leftarrow \text{2nd slope estimate} \\ y_{n+1} = y_n + h \cdot \underbrace{\tfrac{1}{2}(k_1 + k_2)}_{\text{average of the slope estimates}} & \end{cases}$$



$k_2 = f(x_{n+1}, u_{n+1})$

Improved Euler has smaller error
HW 12 — asks you to code this

NOTE: Written HW — Print out / copy paste your code & submit it.

Summary:

Type	Eqn	Method
Euler's Method	$y_{n+1} = y_n + h\,f(x_n, y_n)$	code
Improved Euler's Method	$k_1 = f(x_n, y_n)$ $u_{n+1} = y_n + h k_1$ $k_2 = f(x_{n+1}, u_{n+1})$ $y_{n+1} = y_n + h \cdot \frac{1}{2}(k_1 + k_2)$	code