Quantifying how fast the convergence is, in various root-finding algorithms

Order of convergence.

If you have a sequence (x_k) converging to a root x_*, $e_k = x_* - x_k$ = error in the k^{th} iteration.

We want $e_k \to 0$ quickly.

If $\dfrac{|e_{k+1}|}{|e_k|^d}$ is bounded for all k,

then we say that the convergence has order $\geq d$.

(The order = largest possible d that works.)

$$= \sup \left\{ d : \frac{|e_{k+1}|}{|e_k|^d} \text{ is bounded for all } k \right\}.$$

Faster convergence $\Longleftrightarrow$ larger values of d.

For the bisection method, typically $|e_{k+1}| \approx \frac{1}{2}|e_k|$.

Probably $\dfrac{|e_{k+1}|}{|e_k|} \approx \frac{1}{2}$ (converges to $\frac{1}{2}$).

→ for bisection method:
order of convergence is 1.

→ We explore with sagemath.

Why does Newton's method have order 2 or higher if the root is simple ($(x-3)$) (not multiple root $(x-3)^2$)?

Ans. $f(x)=0$ Taylor poly with remainder for $f(x)$ centered at x_n, evaluated at $x_* =$ exact root.

$$f(x_*) = f(x_n) + f'(x_n)(x_* - x_n) + \frac{f''(c)}{2!}(x_* - x_n)^2$$

since x_* is a root of $f(x)=0$, c is between x_* & x_n.

$$0 = f(x_n) + f'(x_n)\underbrace{(x_* - x_n)}_{e_n} + \frac{1}{2}f''(c)(x_* - x_n)^2$$

Divide by $f'(x_n)$ (assuming it's not zero → always true for a simple root)

$$0 = \frac{f(x_n)}{f'(x_n)} + x_* - x_n + \frac{1}{2}\frac{f''(c)}{f'(x_n)}(x_* - x_n)^2$$

$$-x_* + x_n - \frac{f(x_n)}{f'(x_n)} = \frac{1}{2}\frac{f''(c)}{f'(x_n)} e_n^2$$

$$x_{n+1}$$

$$-e_{n+1} = \frac{1}{2}\frac{f''(c)}{f'(x_n)} e_n^2$$

$$\frac{e_{n+1}}{e_n^2} = -\frac{1}{2}\frac{f''(c)}{f'(x_n)}$$

converges to

$$-\frac{1}{2}\frac{f''(x_*)}{f'(x_*)} = A$$

$$\frac{e_{n+1}}{e_n^2} \to A \quad \text{as } n \to \infty.$$

∴ Newton-Raphson has order 2 convergence!

Tips in Python:

- Sometimes, when an error occurs in Spyder, a system file will appear on the left — don't change it, just close it.
- Global variables

```
b = 5
def ff(x):
    ~~
    ~~
    b = 6
```

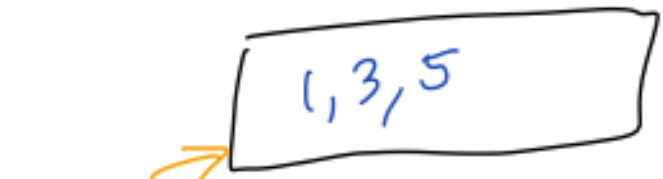
b will not change based on what happens in ff.

Alternative

```
b = 5
def ff(x):
    global b
    ~~
    ~~
    b = 6
```

Now b will change on the outside.

With tkinter Entry widgets:



← user puts something in here.

txtBubba = tk.Entry(----)

In your code, you want to use the user's info.

s = txtBubba.get()

s is always a string.

s = "1,3,5" ← what we want is [1,3,5].

slist = list(eval(s))

takes string and put ( ) around it.