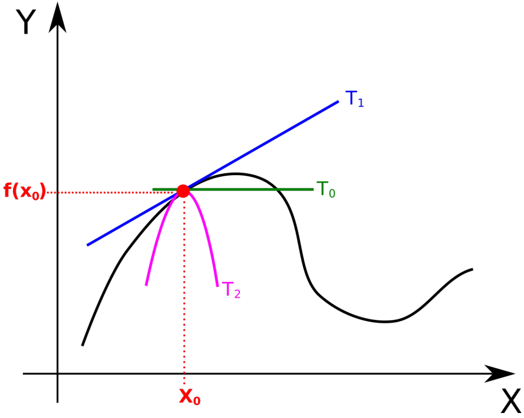
CFD 0 - Exercise 02

Taylor-Series

$$
\begin{aligned}
T_N(x; x_0) &= \sum_{n=0}^{N} \frac{1}{n!} f^{(n)}(x_0) \cdot (x - x_0)^n \\
&= f(x_0) + \frac{1}{1!} f'(x_0) \cdot (x - x_0)^1 \\
&\quad + \frac{1}{2!} f''(x_0) \cdot (x - x_0)^2 \\
&\quad + \dots \\
&\quad + \frac{1}{N!} f^{(N)}(x_0) \cdot (x - x_0)^N
\end{aligned}
$$



$x_0 \rightarrow$ Point for series expansion.

$f(x_0), f'(x_0), f''(x_0), \dots, f^{(N)}(x_0) \rightarrow$ Known values of the function and its derivatives at x_0.

CFD 0 - Exercise 02

Taylor-Series

Example: $f(x) = e^{-x}$ at $x_0 = 0$

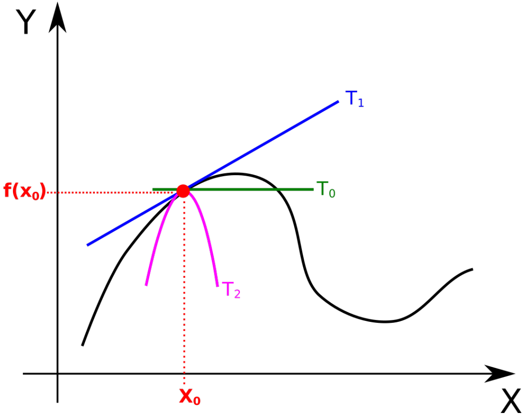
$$\begin{aligned} f(x) &= +e^{-x} &\implies f(0) &= +1 \\ f'(x) &= -e^{-x} &\implies f'(0) &= -1 \\ f''(x) &= +e^{-x} &\implies f''(0) &= +1 \end{aligned}$$

$$\vdots$$

Series up to 2nd order, $T_2(x;0)$:

```python
def taylor2(x):
    return 1. - x + 0.5 * x**2

x = np.linspace(a,b,n_pts)
y = taylor2(x)
```



CFD 0 - Exercise 02

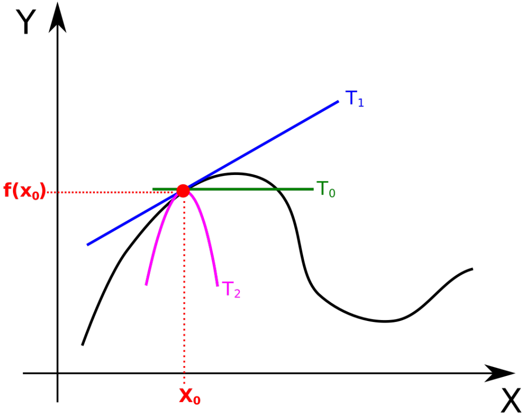
Taylor-Series

Example: $f(x) = e^{-x}$, at $x_0 = 0$

For the general case, $T_N(x; 0)$:

```
def taylorN(x,N):
    y = np.zeros_like(x)
    for n in np.arange(N+1):
        y += (-x)**n / float(fac(n))
    return y

x = np.linspace(a,b,n_pts)
y = taylorN(x)
```



`zeros_like` → zeros-vector with same length as x

`arange` → all numbers in range 0,1,2,...N

`+=` → increment, same like `y = y + ...`

`float` → for floating-point division