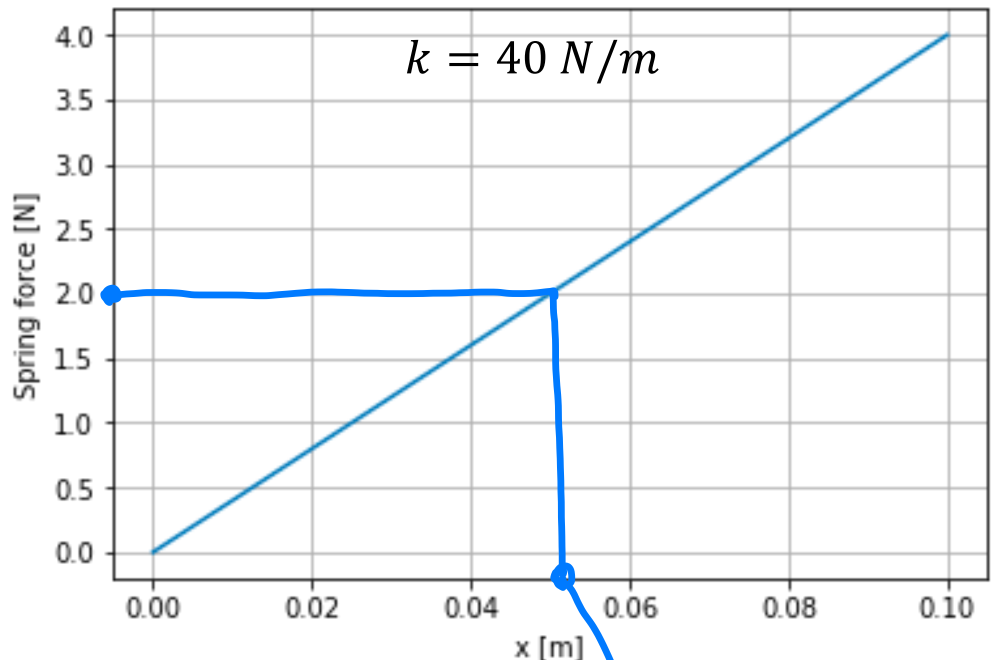
# Nonlinear Equations

# How can we solve these equations?

- Spring force:

$F = k\,x$

What is the displacement when $F = 2\text{N}$?




$$F = kx$$

$$x = \frac{F}{k} = \frac{2N}{40N/m} = 0.05\,m$$

$x \cong 0.05m$

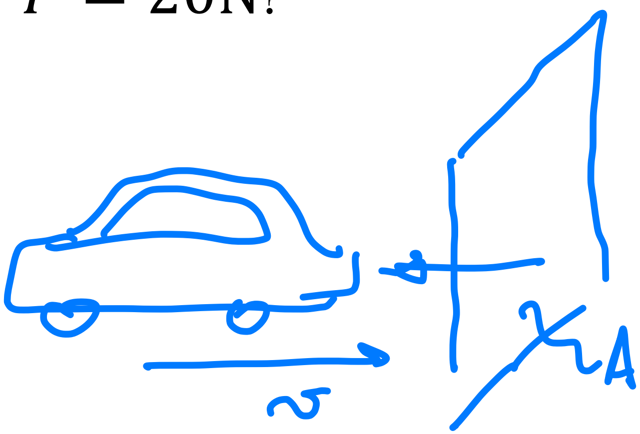
# How can we solve these equations?

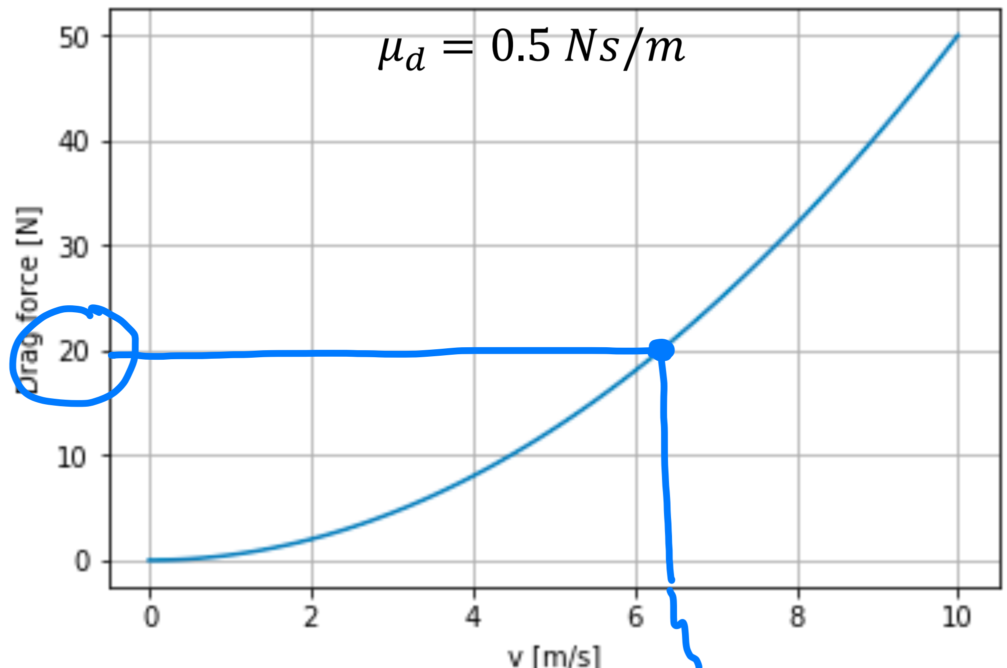
- Drag force:

$$F = 0.5\, C_d\, \rho\, A\, v^2 = \mu_d\;\, v^2$$

What is the velocity when

$F = 20\text{N}$?




$v \cong 6.5\ m/s$

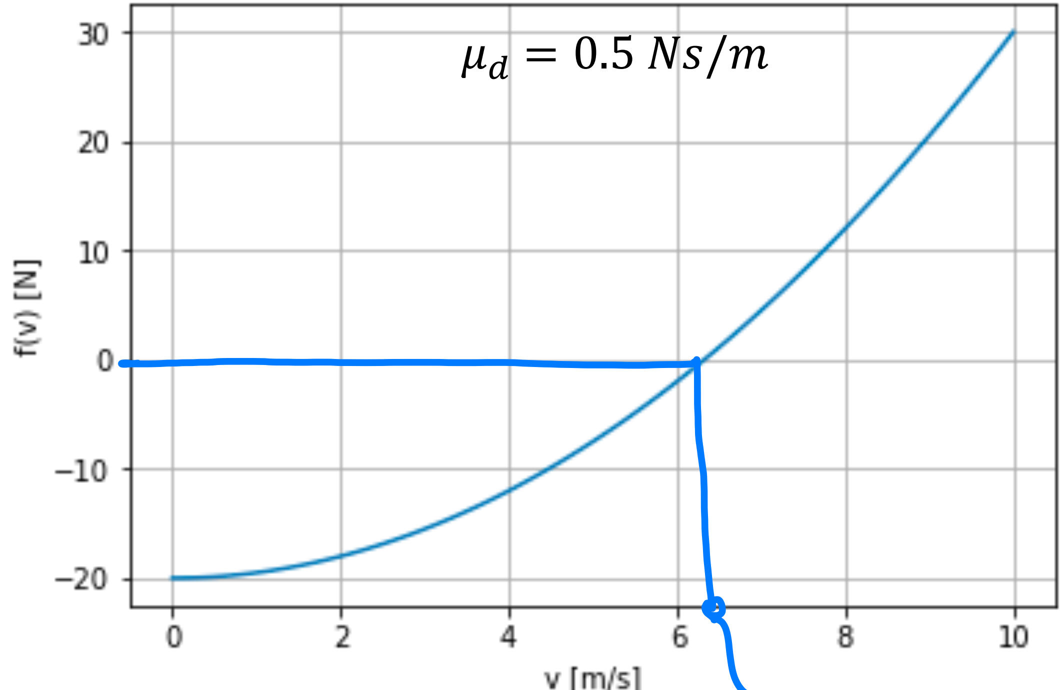
$$F = \mu v^2 \longrightarrow v^2 = \frac{F}{\mu} \longrightarrow v = \pm\sqrt{F/\mu} \Rightarrow v \cong 6.3\ m/s$$

$$F = \mu v^2 \rightarrow \underbrace{F - \mu v^2}_{f(v)} = 0$$

$$f(v) = 0$$

$$f(v) = \mu_d \; v^2 - F = 0$$

Find the root (zero) of the nonlinear equation $f(v)$




$v \cong 6.3\ m/s$

$$\boxed{f(v) = 0}$$

root of $f$

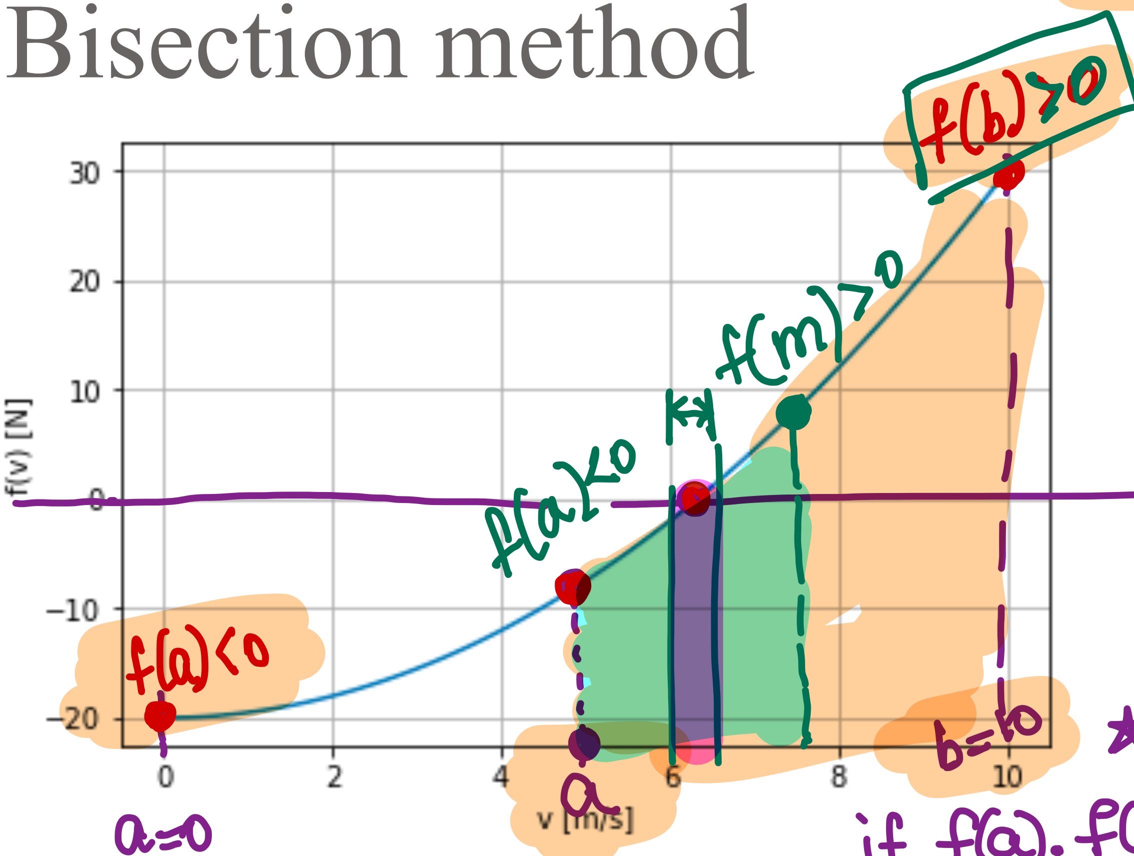
numerical → iterative

# Nonlinear Equations in 1D

**Goal:** Solve $f(x) = 0$ for $f: \mathcal{R} \to \mathcal{R}$

Often called **Root Finding**

# Bisection method




★ Define interval that has the root

$\boxed{\text{interval} : [a, b]}$

$a = 0 \; ; \; b = 10$

$t_0 = |b-a| = 10$

★ Define midpoint:

$m = \frac{b+a}{2} = 5$

★ Check the signs!

check interval:

$sign(f(a)) \neq sign(f(b))$

$f(a) \cdot f(b) < 0$

if $f(a) \cdot f(m) > 0$:

$t_{new} = [m, b]$

$a = m$

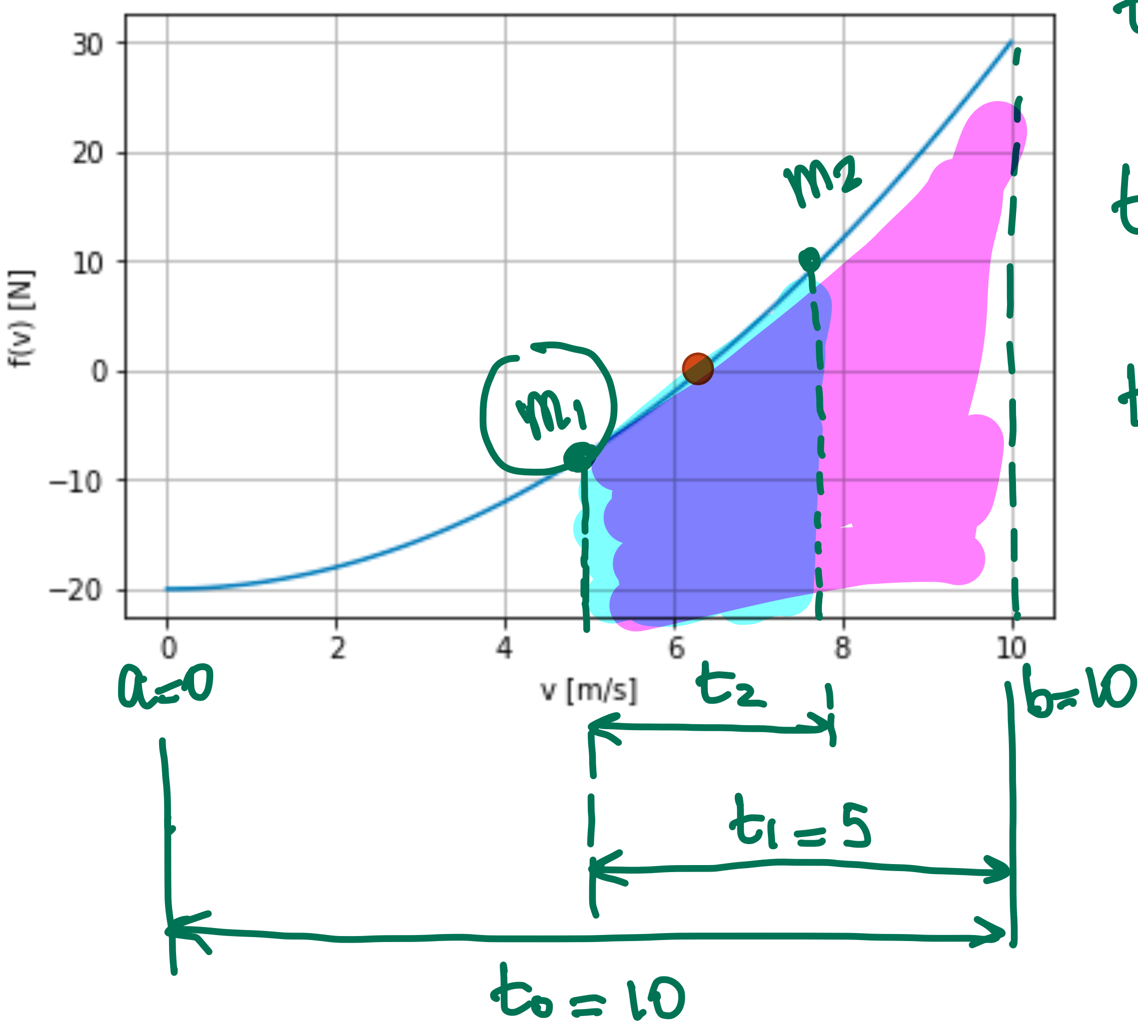
else:

$t_{new} = [a, m]$

$b = m$

interval gets smaller

# Bisection method




$$t_0 = |b-a| = 10$$

$$t_1 = \frac{|b-a|}{2} = \frac{t_0}{2}$$

$$t_2 = \frac{t_1}{2} = \frac{t_0}{2.2}$$

$$t_3 = \frac{t_2}{2} = \frac{t_0}{8}$$

$$\vdots$$

$$\boxed{t_k = \frac{t_0}{2^k}}$$

★ every iteration, the interval is divided by 2 !

# Convergence

An iterative method **converges with rate** $r$ if:

$$\lim_{k\to\infty} \frac{||e_{k+1}||}{||e_k||^r} = C, \qquad 0 < C < \infty \qquad r = 1\text{: linear convergence}$$

Linear convergence gains a constant number of accurate digits each step (and $C < 1$ matters!)

For example: Power Iteration

$$\lim_{k\to\infty} \frac{||e_{k+1}||}{||e_k||^{r=1}} = \boxed{\left|\frac{\lambda_2}{\lambda_1}\right|} = \text{constant} = C \Rightarrow \text{linear convergence}$$

$\star\ \lambda_2 \sim \lambda_1 \to \text{constant} \approx 1 \to$ slow convergence

$\lambda_1 = \alpha\lambda_2 \longrightarrow C = \frac{1}{\alpha} \longrightarrow$ faster convergence as $\alpha$ increases

# Convergence

An iterative method **converges with rate** $r$ if:

Power method

$$\lim_{k\to\infty} \frac{||e_{k+1}||}{||e_k||^r} = C, \qquad 0 < C < \infty$$

- $r = 1$: linear convergence
- $r > 1$: superlinear convergence $1 < r < 2$
- $r = 2$: quadratic convergence

Linear convergence gains a constant number of accurate digits each step (and $C < 1$ matters!)

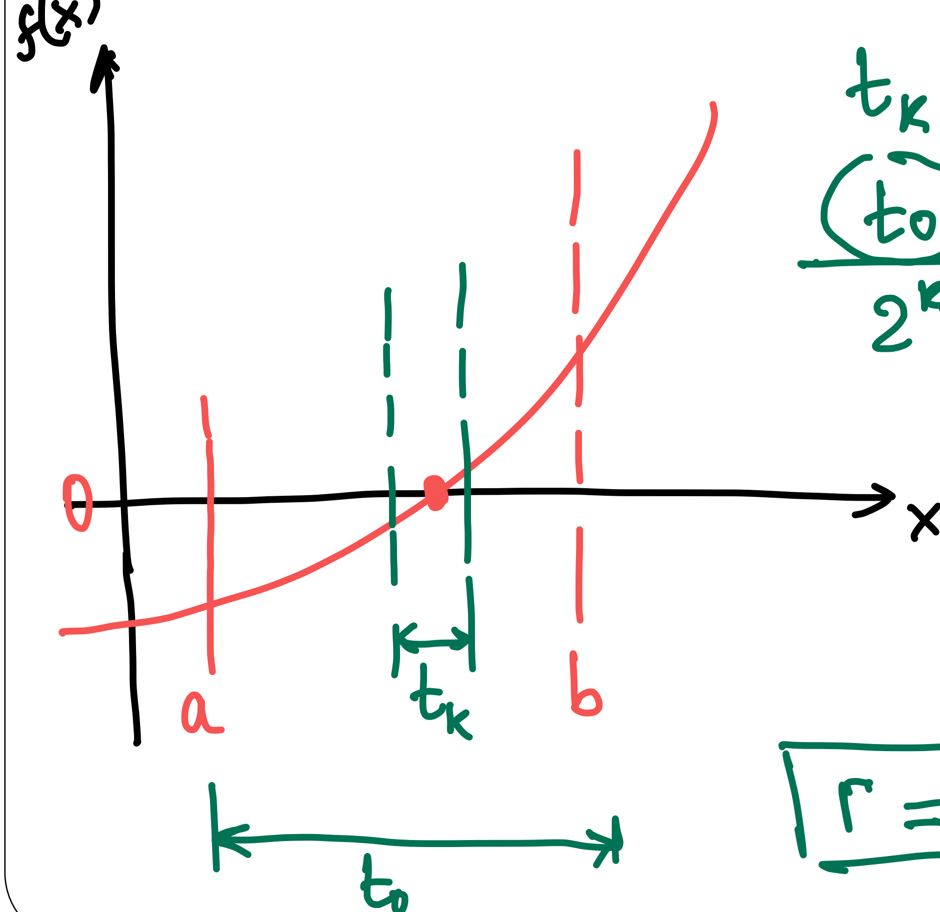
Quadratic convergence doubles the number of accurate digits in each step (however it only starts making sense once $||e_k||$ is small (and $C$ does not matter much)

# Convergence

$x^*$ is the root

$\cancel{x_k} \longrightarrow t_k$ ~~error $= x^* - x_k$~~

- The bisection method does not estimate $x_k$, the approximation of the desired root $x$. It instead finds an interval smaller than a given tolerance that contains the root.




$t_k < tol \longrightarrow$ stop

$$\frac{t_0}{2^k} < tol$$

error $= t_k$

$$\frac{|e_{k+1}|}{|e_k|^r} = \frac{|b-a|/2^{k+1}}{|b-a|/2^k} = \frac{1}{2} = C$$

$\boxed{r=1}$ $\boxed{C=\frac{1}{2}}$ $\Rightarrow$ linear convergence!

# Example:

in general: $t_k < tol$

$$\frac{|b-a|}{2^k} < tol$$

$$2^k > \frac{|b-a|}{tol}$$

$$k > \log_2\left(\frac{|b-a|}{tol}\right)$$

Consider the nonlinear equation

$$f(x) = 0.5x^2 - 2$$

and solving $f(x) = 0$ using the Bisection Method. For each of the initial intervals below, how many iterations are required to ensure the root is accurate within $2^{-4}$?

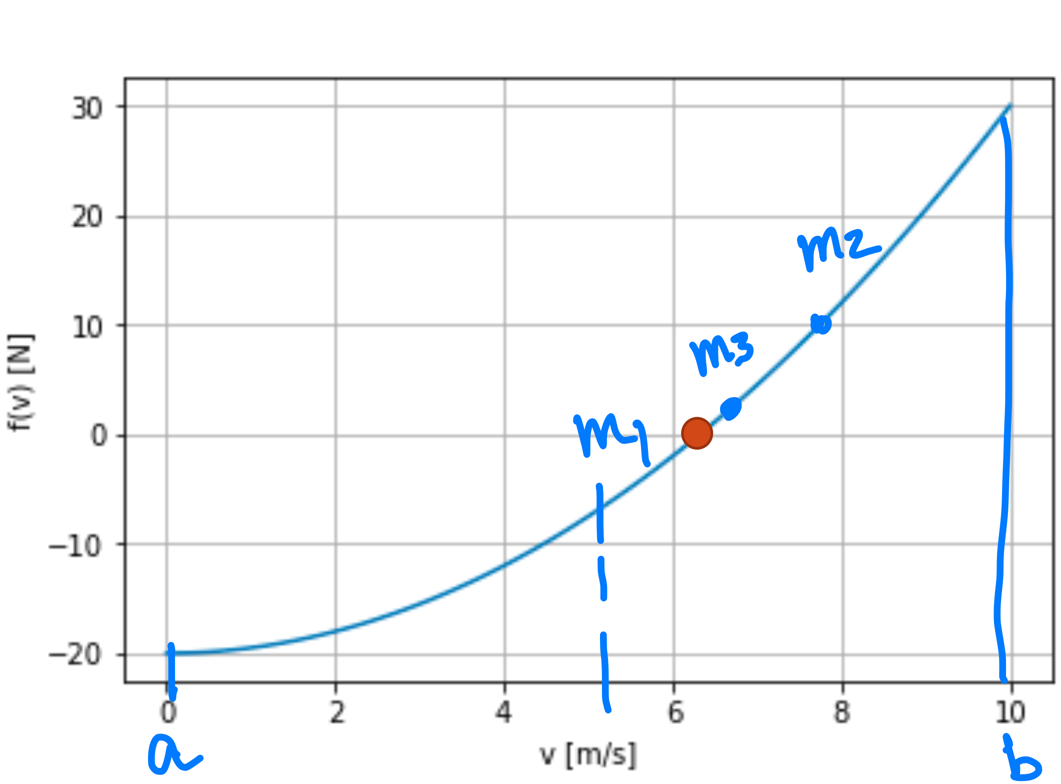
$f(a)$ $f(b)$

*A)* $[-10, -1.8]$ $f(a) \cdot f(b) < 0 \rightarrow$ ok! $\rightarrow k > \log_2\left(\frac{8.2}{2^{-4}}\right) \simeq 7.3$ (8 iterations)

~~*B)* $[-3, -2.1]$~~ $f(a) \cdot f(b) > 0 \rightarrow$ not ok!

*C)* $[-4, 1.9]$ $f(a) \cdot f(b) < 0 \rightarrow$ ok! $\rightarrow k > \log_2\left(\frac{5.9}{2^{-4}}\right) \simeq 6.56$ (7 iterations)

# Bisection method




f(a) , f(b) , f(m)

new fc evaluation

f(m)

**Algorithm:**

1.Take two points, $a$ and $b$, on each side of the root such that $f(a)$ and $f(b)$ have opposite signs.

2.Calculate the midpoint $m = \frac{a+b}{2}$

3. Evaluate $f(m)$ and use $m$ to replace either $a$ or $b$, keeping the signs of the endpoints opposite.

# Bisection Method - summary

- The function must be continuous with a root in the interval $[a, b]$
- Requires only one function evaluations for each iteration!
  - The first iteration requires two function evaluations.
- Given the initial internal $[a, b]$, the length of the interval after $k$ iterations is $\frac{b-a}{2^k}$
- Has linear convergence

# Newton's method

- Recall we want to solve $f(x) = 0$ for $f\colon \mathcal{R} \to \mathcal{R}$
- The Taylor expansion:

$$\underbrace{f(x_k + h)}_{\text{nonlinear}} \approx \underbrace{\underline{f(x_k)} + \underline{f'(x_k)}h}_{\text{linear approximation of } f(x)} = \hat{f}(h)$$

gives a linear approximation for the nonlinear function $f$ near $x_k$.

$$f(x_k + h) = 0$$

$$\hat{f}(h) = 0 \Longrightarrow f(x_k) + f'(x_k)h = 0$$

$$\boxed{h = -\frac{f(x_k)}{f'(x_k)}} \rightarrow \text{Newton step}$$

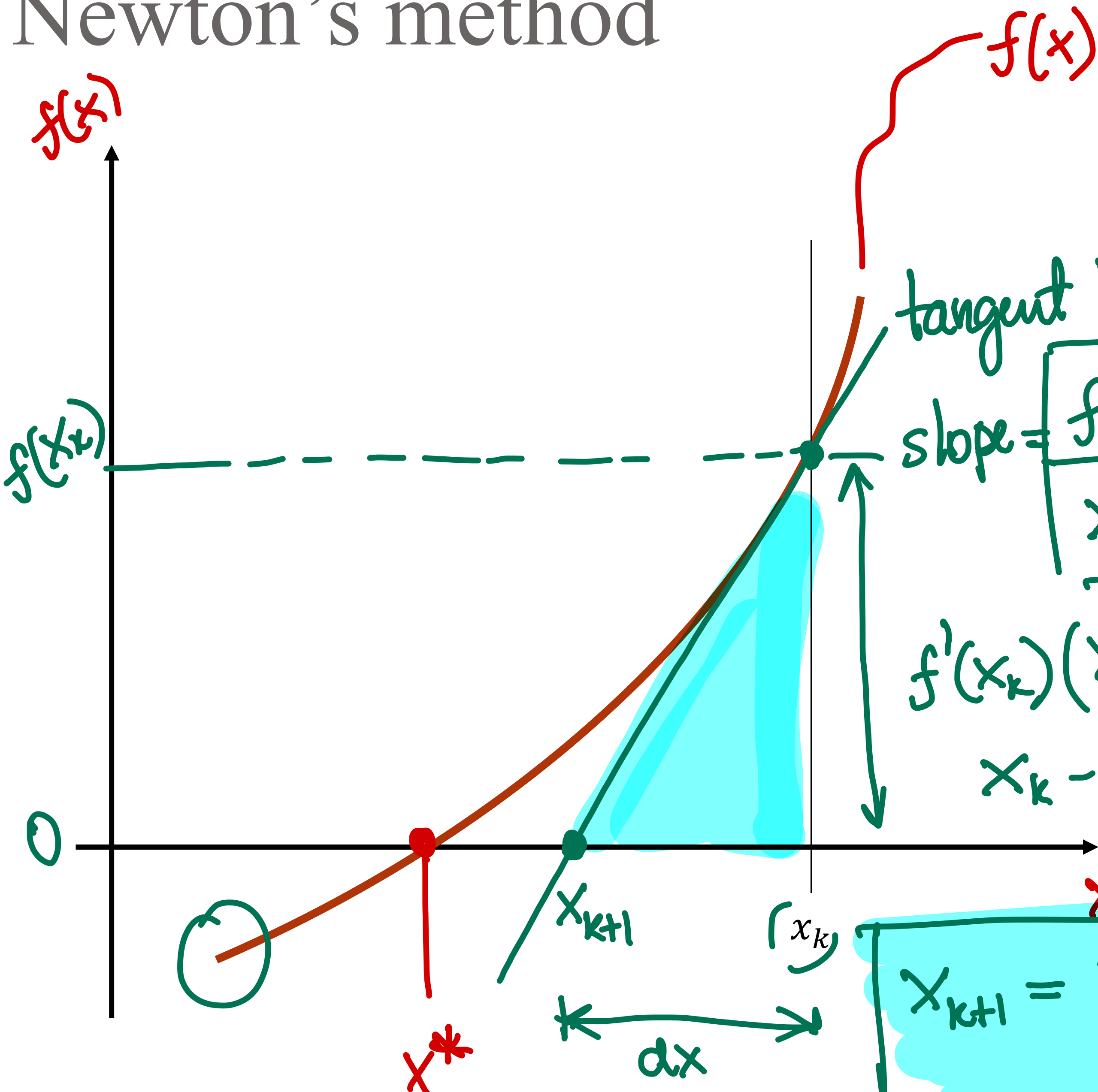
Newton algorithm:

$x_0$ = random (initial guess)

$$f(x_0), f'(x_0) \Rightarrow h \Longrightarrow \boxed{x_{k+1} = x_k + h} \rightarrow \text{Newton update}$$

# Newton's method

Find $x^*$ s.t. $f(x^*) = 0$




tangent line at $x_k$

$$\text{slope} = \boxed{\frac{f(x_k) - 0}{x_k - x_{k+1}} = f'(x_k)}$$

$$f'(x_k)(x_k - x_{k+1}) = f(x_k)$$

$$x_k - x_{k+1} = \frac{f(x_k)}{f'(x_k)}$$

$$\boxed{x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}}$$

# Example

$x_1 = ?$

$x_0 = 0$

Consider solving the nonlinear equation

$$\boxed{5 = 2.0\, e^{x} + x^2} \Rightarrow f(x) = 2e^{x} + x^2 - 5 = 0$$

What is the result of applying **one iteration** of Newton's method for solving nonlinear equations with initial starting guess $x_0 = 0$, i.e. what is $x_1$?

$x_{k+1} = x_k + h \qquad h = -\frac{f(x_k)}{f'(x_k)}$

*A)* $-2$
*B)* $0.75$
*C)* $-1.5$
$\boxed{D)\ 1.5}$
*E)* $3.0$

$f'(x) = 2e^{x} + 2x$

$x_0 \Rightarrow f(x_0) = 2 - 5 = -3 \qquad h = -\frac{f}{f'} = -\frac{(-3)}{2}$

$f'(x_0) = 2 \qquad h = 1.5$

$x_1 = x_0 + h = 0 + 1.5 \Rightarrow \boxed{x_1 = 1.5}$

# Newton's Method - summary

- Must be started with initial guess close enough to root (convergence is only local). Otherwise it may not converge at all.
- Requires function and first derivative evaluation at each iteration (think about two function evaluations)
- Typically has quadratic convergence

$$\lim_{k\to\infty} \boxed{\frac{||e_{k+1}||}{||e_k||^{\boxed{2}}}} = \boxed{C,} \qquad 0 < C < \infty$$

$r=2$

- What can we do when the derivative evaluation is too costly (or difficult to evaluate)?

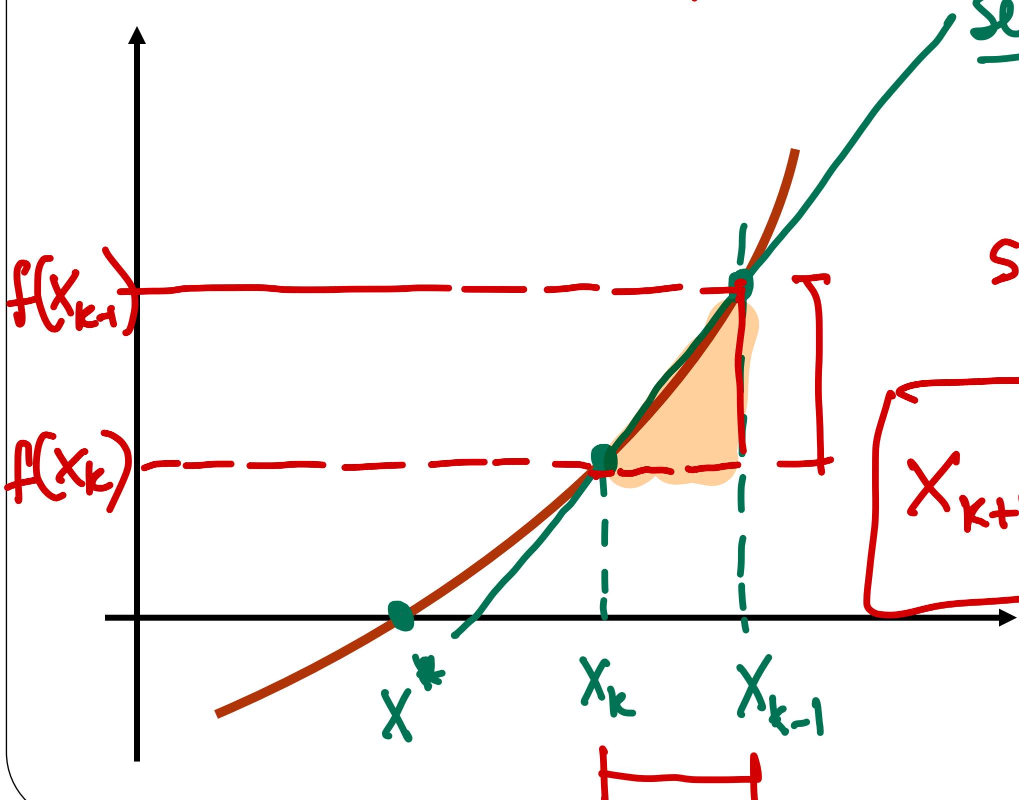
# Secant method

$df \Rightarrow$ approximation for $f'(x)$

Also derived from Taylor expansion, but instead of using $f'(x_k)$, it approximates the tangent with the secant line:

$$x_{k+1} = x_k - f(x_k)/\cancel{f'(x_k)} \longrightarrow x_{k+1} = x_k - \frac{f(x_k)}{df(x_k)}$$

secant line (2 points)! $\boxed{x_0, x_1}$




$$\text{slope} = \frac{f(x_{k-1}) - f(x_k)}{x_{k-1} - x_k} = df$$

$$x_{k+1} = x_k - \frac{f(x_k)}{\frac{f(x_k) - f(x_{k-1})}{(x_k - x_{k-1})}}$$

# Secant Method - summary

- ❑ Still local convergence
- ❑ Requires only one function evaluation per iteration (only the first iteration requires two function evaluations)
- ❑ Needs two starting guesses
- ❑ Has slower convergence than Newton's Method – superlinear convergence

$$\lim_{k\to\infty} \frac{||e_{k+1}||}{||e_k||^r} = C, \qquad 1 < r < 2$$

$f' \longrightarrow df$

# 1D methods for root finding:

| Method | Update | Convergence | Cost |
|---|---|---|---|
| Bisection | Check signs of $f(a)$ and $f(b)$ $$t_k = \frac{\lvert b-a \rvert}{2^k}$$ | Linear ($r = 1$ and $c = 0.5$) | One function evaluation per iteration, no need to compute derivatives |
| Secant | $$x_{k+1} = x_k + h$$ $$h = -f(x_k)/f'(x_k)$$ | Superlinear ($r = 1.618$), local convergence properties, convergence depends on the initial guess | One function evaluation per iteration (two evaluations for the initial guesses only), no need to compute derivatives |
| Newton | $$x_{k+1} = x_k + h$$ $$h = -f(x_k)/dfa$$ $$dfa = \frac{f(x_k) - f(x_{k-1})}{(x_k - x_{k-1})}$$ | Quadratic ($r = 2$), local convergence properties, convergence depends on the initial guess | Two function evaluations per iteration, requires first order derivatives |