

Numerical Methods

ESC101: Fundamentals of Computing

Purushottam Kar

Agenda

- Finding the square root of a number
- Finding the root (zero) of an arbitrary function
- Learn a generalization of the binary search method

Finding Square Roots

The Babylonian Method



The trick of the Babylonians

- Square roots: studied since antiquity – will study a method called the “Babylonian method” – known for over 3000 years!!
- We have a real number $v > 0$ and we wish to find \sqrt{v}
- But \sqrt{v} may be irrational so it may be impossible to represent it using 32 bit floating point numbers ☹️
- Change our goal to find a number r such that $|\sqrt{v} - r| < \epsilon$
- Quantities like ϵ are often called the *tolerance* of the algorithm
- Not unusual to have algorithms that offer $\epsilon \approx 10^{-10}$ or so 😊



The trick of the Babylonians

- Relies on a curious property of the square root
- Suppose we have an estimate $x > 0$ of the true square root \sqrt{v}
 - If x is an *overestimate* i.e. $x > \sqrt{v}$, then $v/x < \sqrt{v}$ - gives us a cute result

For any $x > 0, v > 0$ we always have $\sqrt{v} \in [v/x, x]$
 - A similar result if x is an *underestimate*: if $x < \sqrt{v}$, then $v/x > \sqrt{v}$
- The Babylonian method exploits this to set up an *active region* over the entire positive real line \mathbb{R}_+
- Will maintain invariant that the active region always contains \sqrt{v}
 - The above cute result will help us keep our promise 😊
- Will halve the length of this active region at every step!

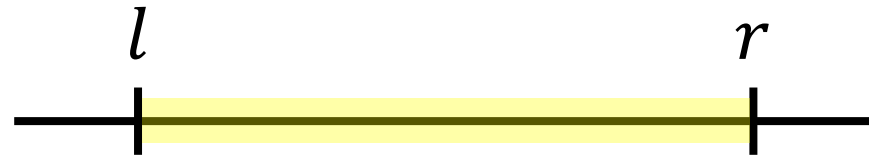
The Babylonian Method

- At every step, active region will look like $[l, r]$ where $l = v/r$ where r is our current estimate of the root
- Start off with an overestimate (if $v > 1$ use $r = v$ else use $r = 1$)
- Update the estimate to $r^{\text{new}} = \frac{1}{2}(l + r)$
- Update the active region to $[l^{\text{new}}, r^{\text{new}}]$ where $l^{\text{new}} = v/r^{\text{new}}$
- Exercise: prove that we have $r \geq r^{\text{new}} \geq \sqrt{v}$ and $l^{\text{new}} \geq l$
- This means that length of active region always shrinks by half 😊



The Babylonian Method

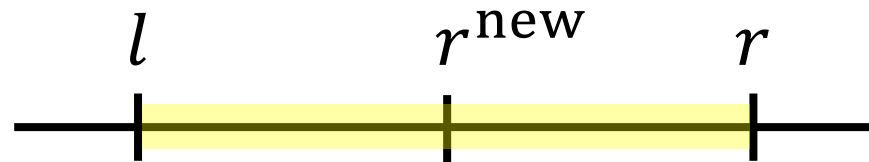
- At every step, active region will look like $[l, r]$ where $l = v/r$ where r is our current estimate of the root
- Start off with an overestimate (if $v > 1$ use $r = v$ else use $r = 1$)
- Update the estimate to $r^{\text{new}} = \frac{1}{2}(l + r)$
- Update the active region to $[l^{\text{new}}, r^{\text{new}}]$ where $l^{\text{new}} = v/r^{\text{new}}$
- Exercise: prove that we have $r \geq r^{\text{new}} \geq \sqrt{v}$ and $l^{\text{new}} \geq l$



- This means that length of active region always shrinks by half 😊

The Babylonian Method

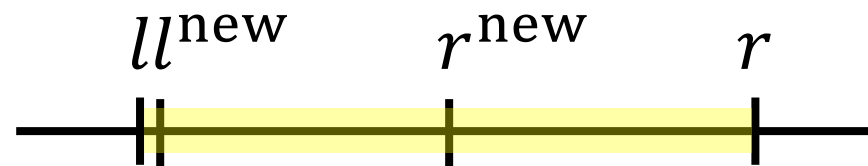
- At every step, active region will look like $[l, r]$ where $l = v/r$ where r is our current estimate of the root
- Start off with an overestimate (if $v > 1$ use $r = v$ else use $r = 1$)
- Update the estimate to $r^{\text{new}} = \frac{1}{2}(l + r)$
- Update the active region to $[l^{\text{new}}, r^{\text{new}}]$ where $l^{\text{new}} = v/r^{\text{new}}$
- Exercise: prove that we have $r \geq r^{\text{new}} \geq \sqrt{v}$ and $l^{\text{new}} \geq l$



- This means that length of active region always shrinks by half ☺

The Babylonian Method

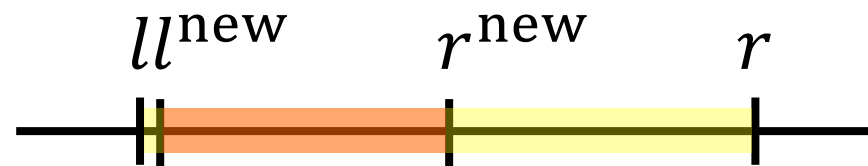
- At every step, active region will look like $[l, r]$ where $l = v/r$ where r is our current estimate of the root
- Start off with an overestimate (if $v > 1$ use $r = v$ else use $r = 1$)
- Update the estimate to $r^{\text{new}} = \frac{1}{2}(l + r)$
- Update the active region to $[l^{\text{new}}, r^{\text{new}}]$ where $l^{\text{new}} = v/r^{\text{new}}$
- Exercise: prove that we have $r \geq r^{\text{new}} \geq \sqrt{v}$ and $l^{\text{new}} \geq l$



- This means that length of active region always shrinks by half 😊

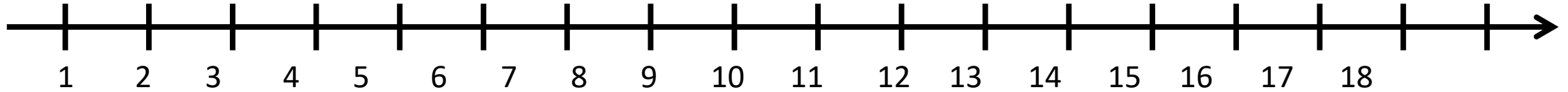
The Babylonian Method

- At every step, active region will look like $[l, r]$ where $l = v/r$ where r is our current estimate of the root
- Start off with an overestimate (if $v > 1$ use $r = v$ else use $r = 1$)
- Update the estimate to $r^{\text{new}} = \frac{1}{2}(l + r)$
- Update the active region to $[l^{\text{new}}, r^{\text{new}}]$ where $l^{\text{new}} = v/r^{\text{new}}$
- Exercise: prove that we have $r \geq r^{\text{new}} \geq \sqrt{v}$ and $l^{\text{new}} \geq l$

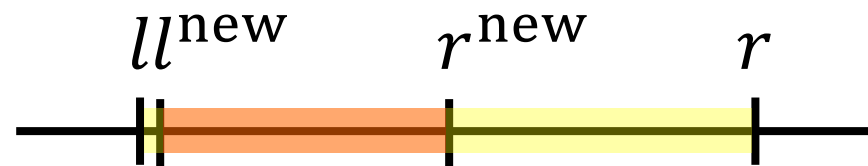


- This means that length of active region always shrinks by half 😊

The Babylonian Method



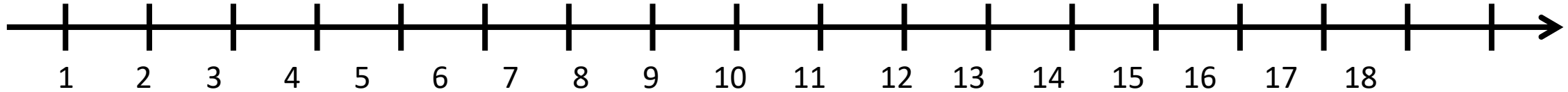
- At every step, active region will look like $[l, r]$ where $l = v/r$ where r is our current estimate of the root
- Start off with an overestimate (if $v > 1$ use $r = v$ else use $r = 1$)
- Update the estimate to $r^{\text{new}} = \frac{1}{2}(l + r)$
- Update the active region to $[l^{\text{new}}, r^{\text{new}}]$ where $l^{\text{new}} = v/r^{\text{new}}$
- Exercise: prove that we have $r \geq r^{\text{new}} \geq \sqrt{v}$ and $l^{\text{new}} \geq l$



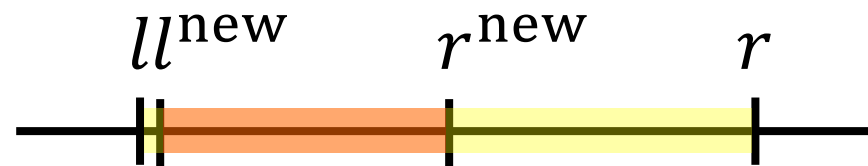
- This means that length of active region always shrinks by half ☺

The Babylonian Method

$$v = 16$$

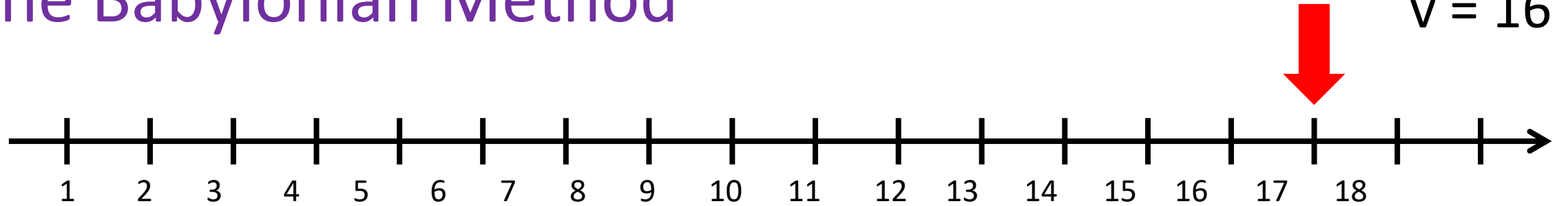


- At every step, active region will look like $[l, r]$ where $l = v/r$ where r is our current estimate of the root
- Start off with an overestimate (if $v > 1$ use $r = v$ else use $r = 1$)
- Update the estimate to $r^{\text{new}} = \frac{1}{2}(l + r)$
- Update the active region to $[l^{\text{new}}, r^{\text{new}}]$ where $l^{\text{new}} = v/r^{\text{new}}$
- Exercise: prove that we have $r \geq r^{\text{new}} \geq \sqrt{v}$ and $l^{\text{new}} \geq l$

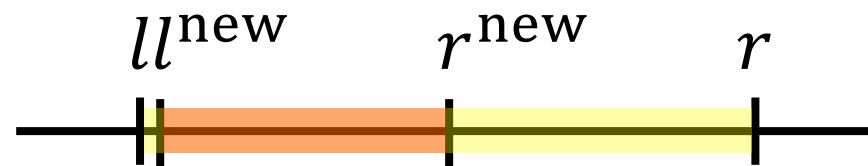


- This means that length of active region always shrinks by half ☺

The Babylonian Method

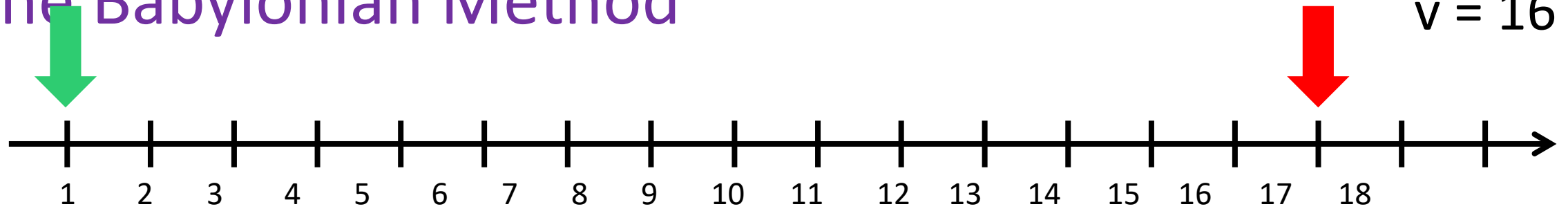


- At every step, active region will look like $[l, r]$ where $l = v/r$ where r is our current estimate of the root
- Start off with an overestimate (if $v > 1$ use $r = v$ else use $r = 1$)
- Update the estimate to $r^{\text{new}} = \frac{1}{2}(l + r)$
- Update the active region to $[l^{\text{new}}, r^{\text{new}}]$ where $l^{\text{new}} = v/r^{\text{new}}$
- Exercise: prove that we have $r \geq r^{\text{new}} \geq \sqrt{v}$ and $l^{\text{new}} \geq l$

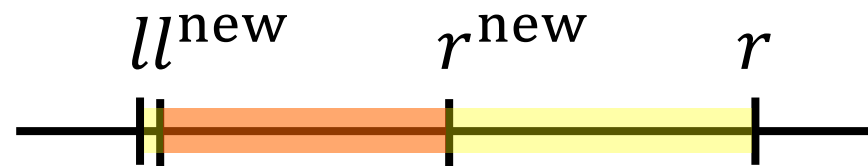


- This means that length of active region always shrinks by half 😊

The Babylonian Method



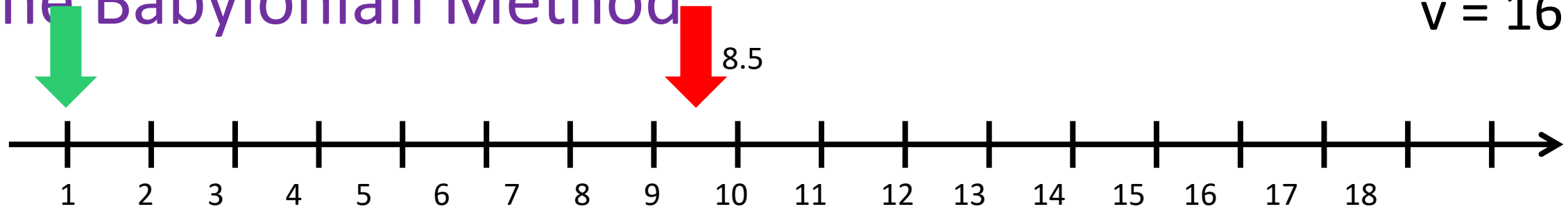
- At every step, active region will look like $[l, r]$ where $l = v/r$ where r is our current estimate of the root
- Start off with an overestimate (if $v > 1$ use $r = v$ else use $r = 1$)
- Update the estimate to $r^{\text{new}} = \frac{1}{2}(l + r)$
- Update the active region to $[l^{\text{new}}, r^{\text{new}}]$ where $l^{\text{new}} = v/r^{\text{new}}$
- Exercise: prove that we have $r \geq r^{\text{new}} \geq \sqrt{v}$ and $l^{\text{new}} \geq l$



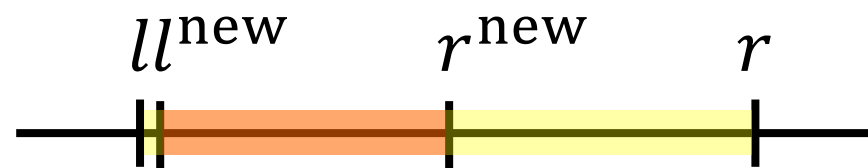
- This means that length of active region always shrinks by half ☺

The Babylonian Method

$$v = 16$$



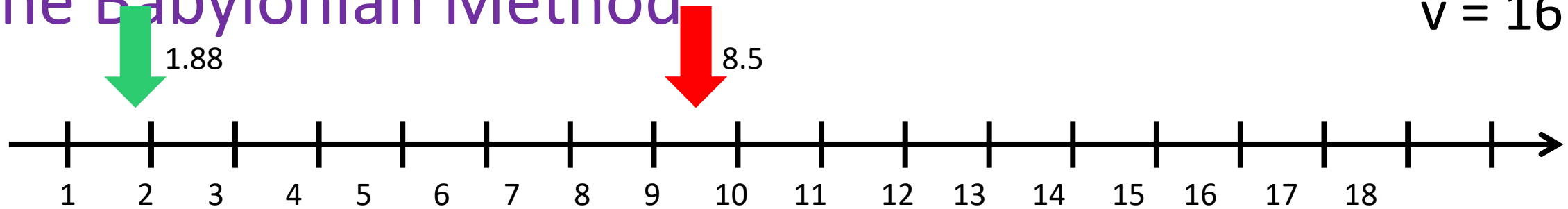
- At every step, active region will look like $[l, r]$ where $l = v/r$ where r is our current estimate of the root
- Start off with an overestimate (if $v > 1$ use $r = v$ else use $r = 1$)
- Update the estimate to $r^{\text{new}} = \frac{1}{2}(l + r)$
- Update the active region to $[l^{\text{new}}, r^{\text{new}}]$ where $l^{\text{new}} = v/r^{\text{new}}$
- Exercise: prove that we have $r \geq r^{\text{new}} \geq \sqrt{v}$ and $l^{\text{new}} \geq l$



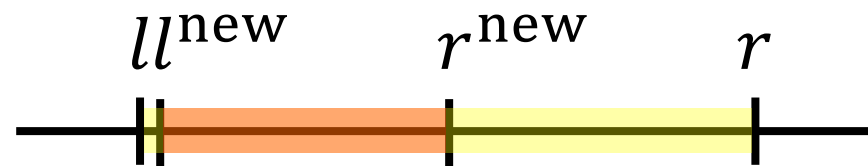
- This means that length of active region always shrinks by half ☺

The Babylonian Method

$$v = 16$$



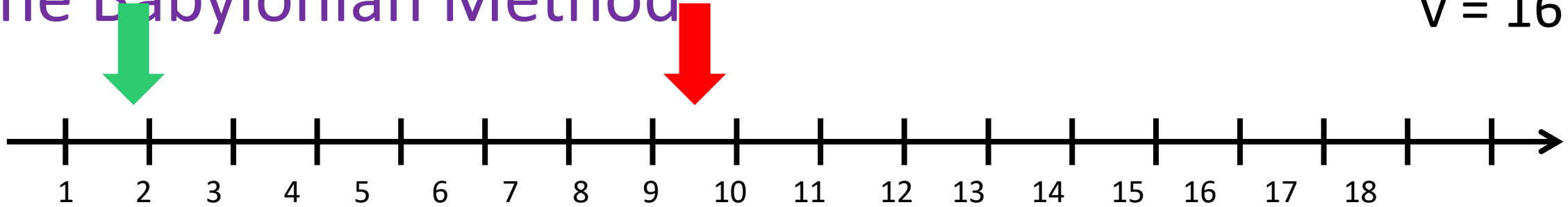
- At every step, active region will look like $[l, r]$ where $l = v/r$ where r is our current estimate of the root
- Start off with an overestimate (if $v > 1$ use $r = v$ else use $r = 1$)
- Update the estimate to $r^{\text{new}} = \frac{1}{2}(l + r)$
- Update the active region to $[l^{\text{new}}, r^{\text{new}}]$ where $l^{\text{new}} = v/r^{\text{new}}$
- Exercise: prove that we have $r \geq r^{\text{new}} \geq \sqrt{v}$ and $l^{\text{new}} \geq l$



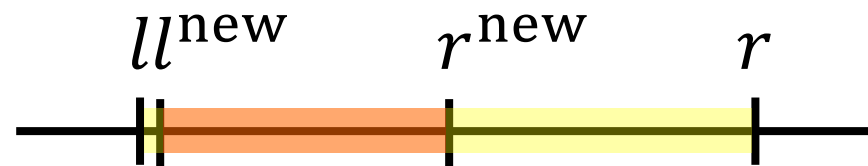
- This means that length of active region always shrinks by half 😊

The Babylonian Method

$$v = 16$$



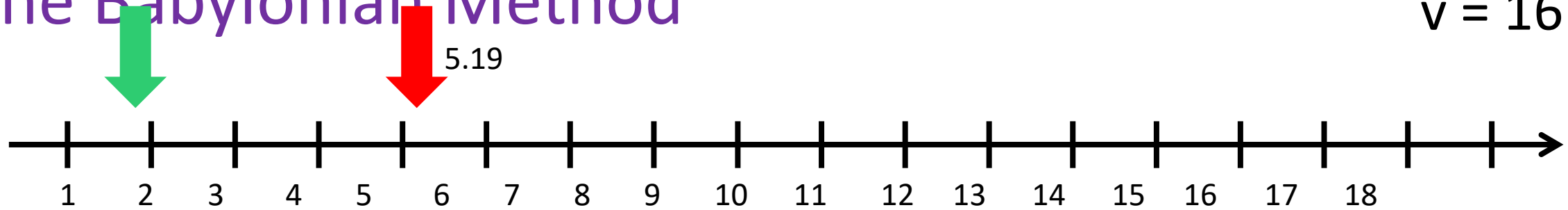
- At every step, active region will look like $[l, r]$ where $l = v/r$ where r is our current estimate of the root
- Start off with an overestimate (if $v > 1$ use $r = v$ else use $r = 1$)
- Update the estimate to $r^{\text{new}} = \frac{1}{2}(l + r)$
- Update the active region to $[l^{\text{new}}, r^{\text{new}}]$ where $l^{\text{new}} = v/r^{\text{new}}$
- Exercise: prove that we have $r \geq r^{\text{new}} \geq \sqrt{v}$ and $l^{\text{new}} \geq l$



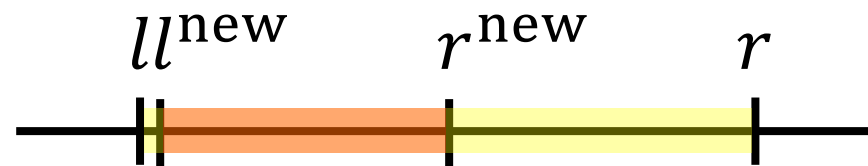
- This means that length of active region always shrinks by half ☺

The Babylonian Method

$$v = 16$$



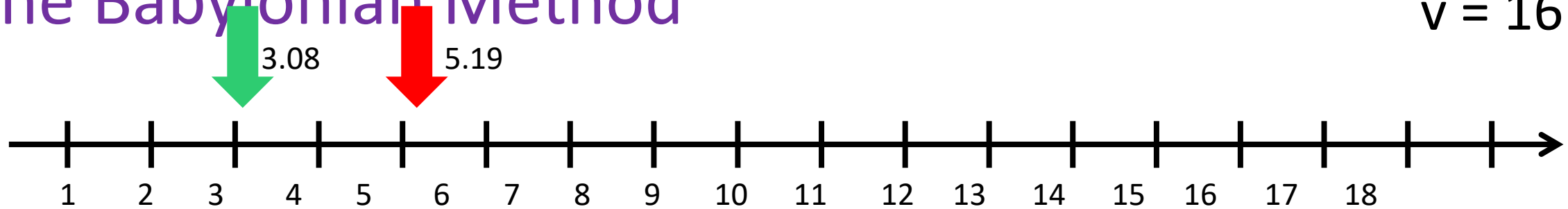
- At every step, active region will look like $[l, r]$ where $l = v/r$ where r is our current estimate of the root
- Start off with an overestimate (if $v > 1$ use $r = v$ else use $r = 1$)
- Update the estimate to $r^{\text{new}} = \frac{1}{2}(l + r)$
- Update the active region to $[l^{\text{new}}, r^{\text{new}}]$ where $l^{\text{new}} = v/r^{\text{new}}$
- Exercise: prove that we have $r \geq r^{\text{new}} \geq \sqrt{v}$ and $l^{\text{new}} \geq l$



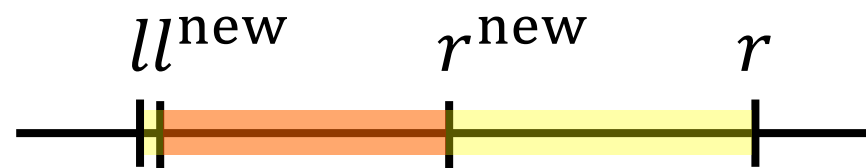
- This means that length of active region always shrinks by half ☺

The Babylonian Method

$$v = 16$$



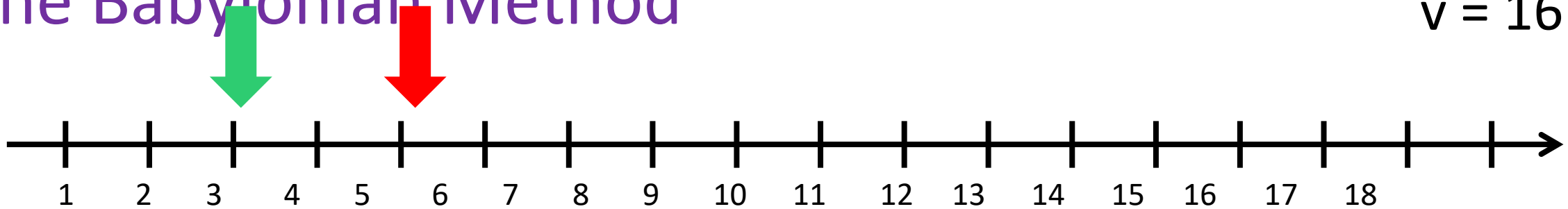
- At every step, active region will look like $[l, r]$ where $l = v/r$ where r is our current estimate of the root
- Start off with an overestimate (if $v > 1$ use $r = v$ else use $r = 1$)
- Update the estimate to $r^{\text{new}} = \frac{1}{2}(l + r)$
- Update the active region to $[l^{\text{new}}, r^{\text{new}}]$ where $l^{\text{new}} = v/r^{\text{new}}$
- Exercise: prove that we have $r \geq r^{\text{new}} \geq \sqrt{v}$ and $l^{\text{new}} \geq l$



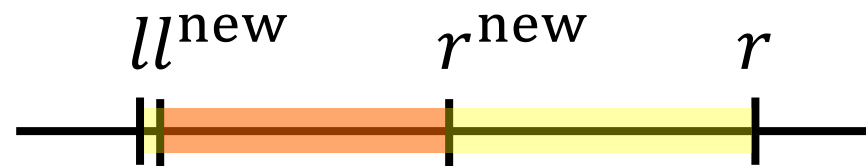
- This means that length of active region always shrinks by half ☺

The Babylonian Method

$$v = 16$$



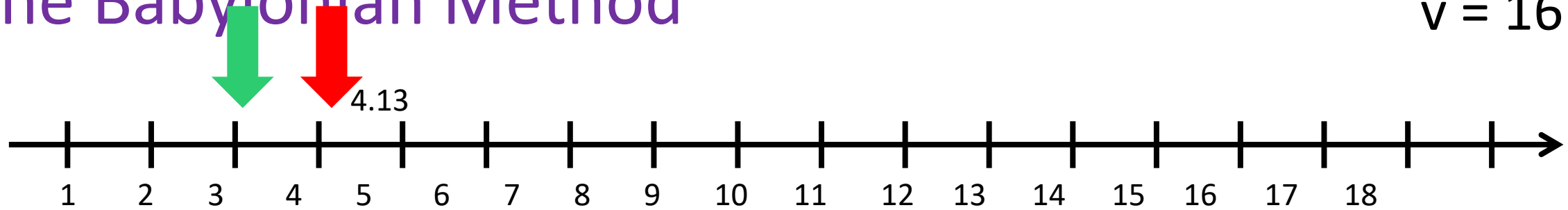
- At every step, active region will look like $[l, r]$ where $l = v/r$ where r is our current estimate of the root
- Start off with an overestimate (if $v > 1$ use $r = v$ else use $r = 1$)
- Update the estimate to $r^{\text{new}} = \frac{1}{2}(l + r)$
- Update the active region to $[l^{\text{new}}, r^{\text{new}}]$ where $l^{\text{new}} = v/r^{\text{new}}$
- Exercise: prove that we have $r \geq r^{\text{new}} \geq \sqrt{v}$ and $l^{\text{new}} \geq l$



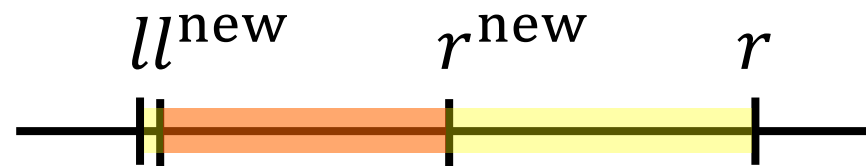
- This means that length of active region always shrinks by half 😊

The Babylonian Method

$$v = 16$$



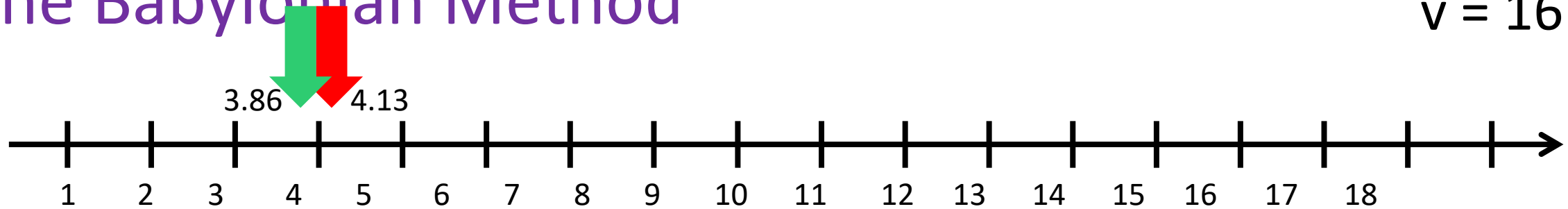
- At every step, active region will look like $[l, r]$ where $l = v/r$ where r is our current estimate of the root
- Start off with an overestimate (if $v > 1$ use $r = v$ else use $r = 1$)
- Update the estimate to $r^{\text{new}} = \frac{1}{2}(l + r)$
- Update the active region to $[l^{\text{new}}, r^{\text{new}}]$ where $l^{\text{new}} = v/r^{\text{new}}$
- Exercise: prove that we have $r \geq r^{\text{new}} \geq \sqrt{v}$ and $l^{\text{new}} \geq l$



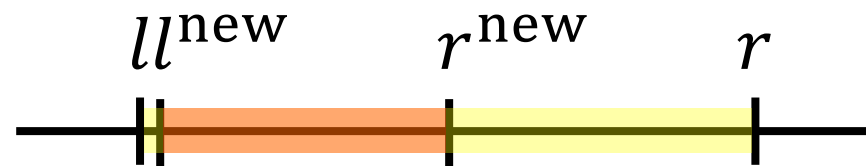
- This means that length of active region always shrinks by half ☺

The Babylonian Method

$$v = 16$$



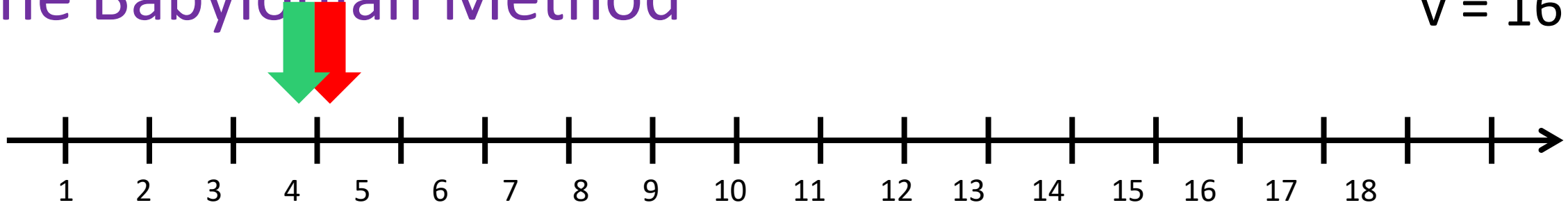
- At every step, active region will look like $[l, r]$ where $l = v/r$ where r is our current estimate of the root
- Start off with an overestimate (if $v > 1$ use $r = v$ else use $r = 1$)
- Update the estimate to $r^{\text{new}} = \frac{1}{2}(l + r)$
- Update the active region to $[l^{\text{new}}, r^{\text{new}}]$ where $l^{\text{new}} = v/r^{\text{new}}$
- Exercise: prove that we have $r \geq r^{\text{new}} \geq \sqrt{v}$ and $l^{\text{new}} \geq l$



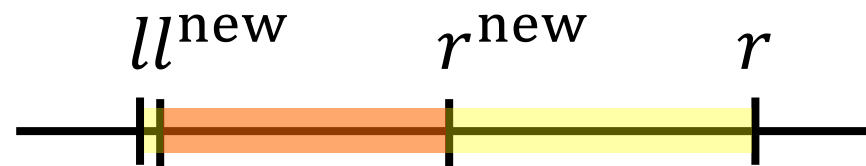
- This means that length of active region always shrinks by half 😊

The Babylonian Method

$$v = 16$$



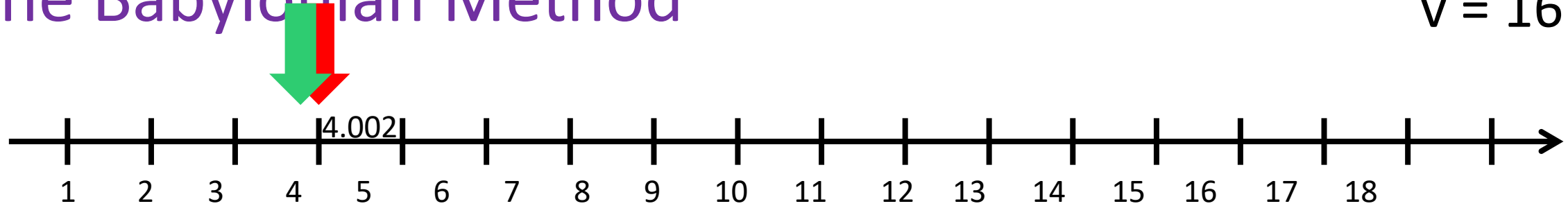
- At every step, active region will look like $[l, r]$ where $l = v/r$ where r is our current estimate of the root
- Start off with an overestimate (if $v > 1$ use $r = v$ else use $r = 1$)
- Update the estimate to $r^{\text{new}} = \frac{1}{2}(l + r)$
- Update the active region to $[l^{\text{new}}, r^{\text{new}}]$ where $l^{\text{new}} = v/r^{\text{new}}$
- Exercise: prove that we have $r \geq r^{\text{new}} \geq \sqrt{v}$ and $l^{\text{new}} \geq l$



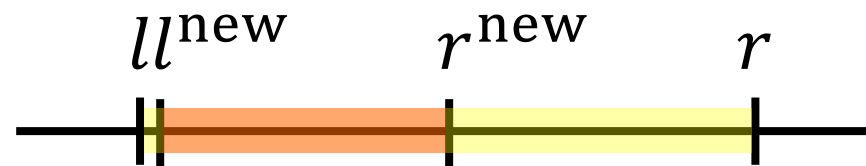
- This means that length of active region always shrinks by half ☺

The Babylonian Method

$$v = 16$$



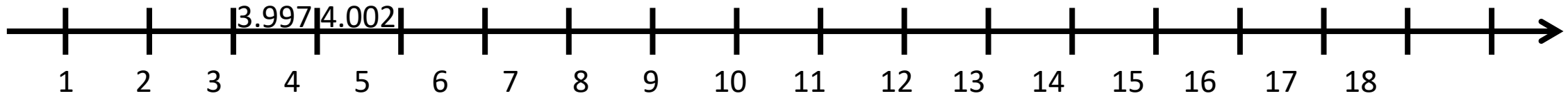
- At every step, active region will look like $[l, r]$ where $l = v/r$ where r is our current estimate of the root
- Start off with an overestimate (if $v > 1$ use $r = v$ else use $r = 1$)
- Update the estimate to $r^{\text{new}} = \frac{1}{2}(l + r)$
- Update the active region to $[l^{\text{new}}, r^{\text{new}}]$ where $l^{\text{new}} = v/r^{\text{new}}$
- Exercise: prove that we have $r \geq r^{\text{new}} \geq \sqrt{v}$ and $l^{\text{new}} \geq l$



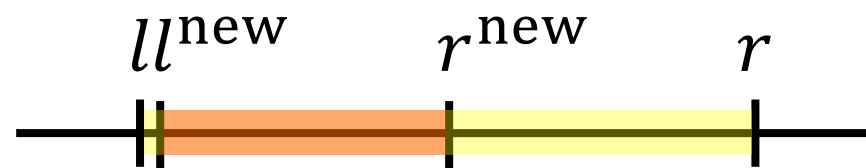
- This means that length of active region always shrinks by half 😊

The Babylonian Method

$$v = 16$$



- At every step, active region will look like $[l, r]$ where $l = v/r$ where r is our current estimate of the root
- Start off with an overestimate (if $v > 1$ use $r = v$ else use $r = 1$)
- Update the estimate to $r^{\text{new}} = \frac{1}{2}(l + r)$
- Update the active region to $[l^{\text{new}}, r^{\text{new}}]$ where $l^{\text{new}} = v/r^{\text{new}}$
- Exercise: prove that we have $r \geq r^{\text{new}} \geq \sqrt{v}$ and $l^{\text{new}} \geq l$



- This means that length of active region always shrinks by half ☺

The Babylonian Method

THE BABYLONIAN METHOD

1. Given: positive number $v > 0$, tolerance $\epsilon > 0$
2. Initialize $r \geq \sqrt{v}$ *//Initial overestimate*
3. Repeat
 1. Let $l \leftarrow v/r$ *//Update left limit of active region*
 2. If $r - l < \epsilon$, return r *//If active region is tiny, we're done*
 3. Set $r \leftarrow \frac{1}{2}(l + r)$ *//Shrink active region by half*

- At every step we are ensured $\sqrt{v} \in [l, r]$. If $r - l < \epsilon$ then we must have $r - \sqrt{v} < \epsilon$ as well as $r \geq \sqrt{v}$ i.e. $|r - \sqrt{v}| < \epsilon$ i.e. we are done!
- As active region halves every time, we will exit loop within $\mathcal{O}\left(\log_2 \frac{1}{\epsilon}\right)$ iterations. For $\epsilon = 10^{-10}$ this means only around 33 iterations 😊



The Babylonian Method

THE BABYLONIAN METHOD

1. Given: positive number $v > 0$, tolerance $\epsilon > 0$
 2. Initialize $r \geq \sqrt{v}$ *//Initial overestimate*
 3. Repeat
 1. Let $l \leftarrow v/r$ *//Update left limit of active region*
 2. If $r - l < \epsilon$, *any, we're done*
 3. Set $r \leftarrow \frac{1}{2}(l + r)$ *the region by half*
- Actually, a more careful analysis can show that the method takes only $\mathcal{O}\left(\log \log \frac{1}{\epsilon}\right)$ iterations – wow!

- At every step we are ensured $\sqrt{v} \in [l, r]$. If $r - l < \epsilon$ then we must have $r - \sqrt{v} < \epsilon$ as well as $r \geq \sqrt{v}$ i.e. $|r - \sqrt{v}| < \epsilon$ i.e. we are done!
- As active region halves every time, we will exit loop within $\mathcal{O}\left(\log_2 \frac{1}{\epsilon}\right)$ iterations. For $\epsilon = 10^{-10}$ this means only around 33 iterations 😊

The Babylonian Method

THE BABYLONIAN METHOD

1. Given: positive number v
 2. Initialize $r \geq \sqrt{v}$ (overestimate)
 3. Repeat
 1. Let $l \leftarrow v/r$ (update left limit of active region)
 2. If $r - l < \epsilon$, $r = \sqrt{v}$ (any, we're done)
 3. Set $r \leftarrow \frac{1}{2}(l + r)$ (shrink active region by half)
- Actually, a more careful analysis can show that the method takes only $\mathcal{O}\left(\log \log \frac{1}{\epsilon}\right)$ iterations – wow!

- At every step we are ensured $\sqrt{v} \in [l, r]$. If $r - l < \epsilon$ then we must have $r - \sqrt{v} < \epsilon$ as well as $r \geq \sqrt{v}$ i.e. $|r - \sqrt{v}| < \epsilon$ i.e. we are done!
- As active region halves every time, we will exit loop within $\mathcal{O}\left(\log_2 \frac{1}{\epsilon}\right)$ iterations. For $\epsilon = 10^{-10}$ this means only around 33 iterations 😊

Finding Roots of Functions

The Bisection Method



The Bisection Method

- Consider a continuous function $f: \mathbb{R} \rightarrow \mathbb{R}$ (say a polynomial) – wish to find its root i.e. some $x_0 \in \mathbb{R}$ such that $f(x_0) = 0$
- Useful in finding eigenvalues of matrices (characteristic poly)
- Useful in optimization algorithms – wait a bit
- Suppose we are given an interval $[a, b]$ so that $f(a) < 0, f(b) > 0$
- Intermediate value theorem: there must lie a root of f in $[a, b]$

The Bisection Method

- Consider a continuous function $f: \mathbb{R} \rightarrow \mathbb{R}$ (say a polynomial) – wish to find its root i.e. some $x_0 \in \mathbb{R}$ such that $f(x_0) = 0$
- Useful in finding eigenvalues of matrices (characteristic poly)
- Useful in optimization algorithms – wait a bit
- Suppose we are given an interval $[a, b]$ so that $f(a) < 0, f(b) > 0$
- Intermediate value theorem: there must lie a root of f in $[a, b]$

Wait – this only tells us that there exist an *odd* number of roots in this interval – there may be more than one root

The Bisection Method

- Consider a continuous function $f: \mathbb{R} \rightarrow \mathbb{R}$ (say a polynomial) – wish to find its root i.e. some $x_0 \in \mathbb{R}$ such that $f(x_0) = 0$
- Useful in finding eigenvalues of matrices (characteristic poly)
- Useful in optimization algorithms – wait a bit
- Suppose we are given an interval $[a, b]$ so that $f(a) < 0, f(b) > 0$
- Intermediate value theorem: there must lie a root of f in $[a, b]$

Wait – this only tells us that there exist an *odd* number of roots in this interval – there may be more than one root

Okay so suppose we want to find any one of those many roots 😊

The Bisection Method

- Consider a continuous function $f: \mathbb{R} \rightarrow \mathbb{R}$ (say a polynomial) – wish to find its root i.e. some $x_0 \in \mathbb{R}$ such that $f(x_0) = 0$
- Useful in finding eigenvalues of matrices (characteristic poly)
- Useful in optimization algorithms – wait a bit
- Suppose we are given an interval $[a, b]$ so that $f(a) < 0, f(b) > 0$
- Intermediate value theorem: there must lie a root of f in $[a, b]$

The Bisection Method

- Consider a continuous function $f: \mathbb{R} \rightarrow \mathbb{R}$ (say a polynomial) – wish to find its root i.e. some $x_0 \in \mathbb{R}$ such that $f(x_0) = 0$
- Useful in finding eigenvalues of matrices (characteristic poly)
- Useful in optimization algorithms – wait a bit
- Suppose we are given an interval $[a, b]$ so that $f(a) < 0, f(b) > 0$
- Intermediate value theorem: there must lie a root of f in $[a, b]$
- Can we find that root? Or a good approximation to it?
- The bisection method does this by generalizing binary search
- Maintains an active region and is careful that the region always contains at least one root
- Will halve the size of that region that each step!



The Bisection Method

- The secret sauce of bisection method is the intermediate value theorem combined with the binary search intuition
- Suppose we are ensured that $f(a) < 0, f(b) > 0$
 - Think about what you would do if instead $f(a) > 0, f(b) < 0$
- Suppose we calculate $f(c)$ for some $c \in (a, b)$ - three cases
 - $f(c) = 0$ Yay – we have found the root – go home and rest!
 - $f(c) < 0$ Apply IVT to $[c, b]$ - there must lie a root in the interval $[c, b]$

The Bisection Method

- The secret sauce of bisection method is the intermediate value theorem
- Suppose $f(a) > 0$ and $f(b) < 0$
 - Then $[a, b]$ contains an odd number of roots and $[a, c]$ contains an even (possibly 0) number of roots
- Suppose we calculate $f(c)$ for some $c \in (a, b)$ three cases
 - $f(c) = 0$ Yay – we have found the root – go home and rest!
 - $f(c) < 0$ Apply IVT to $[c, b]$ - there must lie a root in the interval $[c, b]$



The Bisection Method

- The secret sauce of bisection method is the intermediate value theorem combined with the binary search intuition
- Suppose we are ensured that $f(a) < 0, f(b) > 0$
 - Think about what you would do if instead $f(a) > 0, f(b) < 0$
- Suppose we calculate $f(c)$ for some $c \in (a, b)$ - three cases
 - $f(c) = 0$ Yay – we have found the root – go home and rest!
 - $f(c) < 0$ Apply IVT to $[c, b]$ - there must lie a root in the interval $[c, b]$

The Bisection Method

- The secret sauce of bisection method is the intermediate value theorem combined with the binary search intuition
- Suppose we are ensured that $f(a) < 0, f(b) > 0$
 - Think about what you would do if instead $f(a) > 0, f(b) < 0$
- Suppose we calculate $f(c)$ for some $c \in (a, b)$ - three cases
 - $f(c) = 0$ Yay – we have found the root – go home and rest!
 - $f(c) < 0$ Apply IVT to $[c, b]$ - there must lie a root in the interval $[c, b]$
 - $f(c) > 0$ Apply IVT to $[a, c]$ - there must lie a root in the interval $[a, c]$
- If we choose $c = \frac{1}{2}(a + b)$ then no matter what the case we will halve the active region or else discover a root
- Once active region is tiny, we have found an approximate root



The Bisection Method

THE BISECTION METHOD

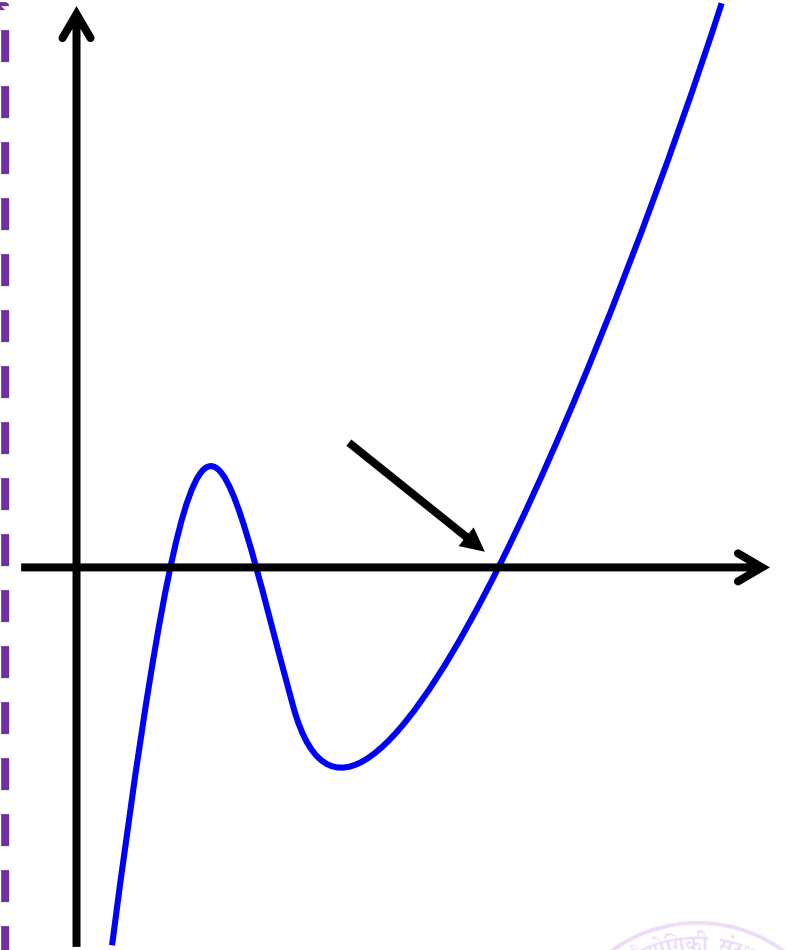
1. Given: continuous function f , active region $[a, b]$, tolerance $\epsilon > 0$
2. Check $f(a) \cdot f(b) < 0$ *//To apply IVT*
3. Repeat
 1. Let $c = \frac{1}{2}(a + b)$ *//Mid point*
 2. If $b - a < \epsilon$ return c *//Approx root*
 3. If $f(c) = 0$ return c *//Found it!*
 4. If $f(c) \cdot f(a) < 0$, $b \leftarrow c$
 5. If $f(c) \cdot f(b) < 0$, $a \leftarrow c$



The Bisection Method

THE BISECTION METHOD

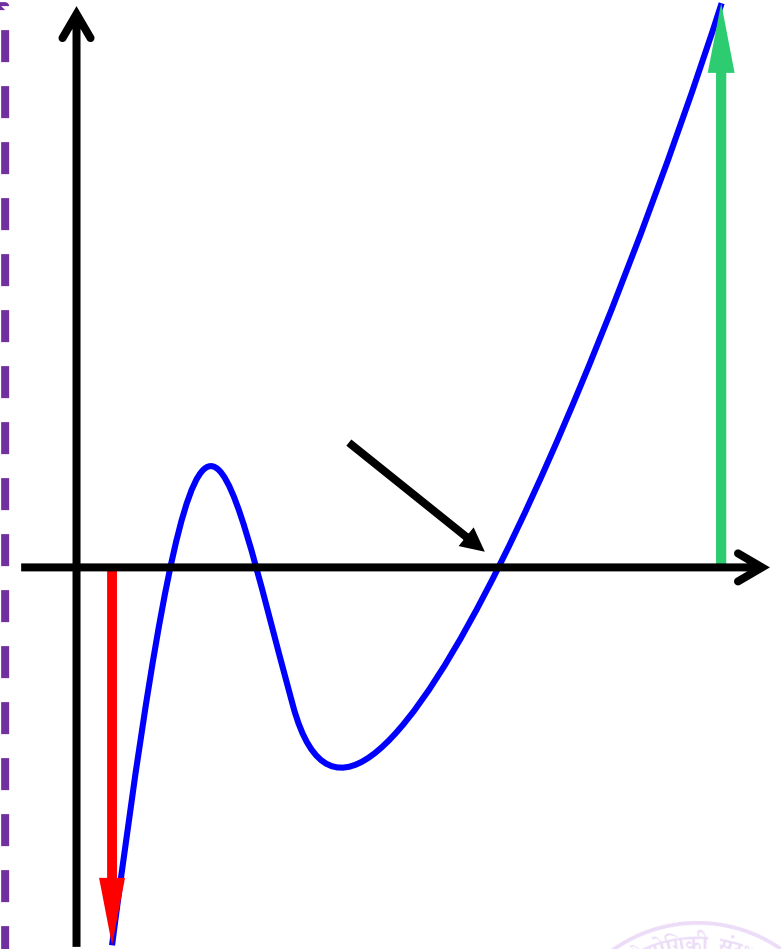
1. Given: continuous function f , active region $[a, b]$, tolerance $\epsilon > 0$
2. Check $f(a) \cdot f(b) < 0$ *//To apply IVT*
3. Repeat
 1. Let $c = \frac{1}{2}(a + b)$ *//Mid point*
 2. If $b - a < \epsilon$ return c *//Approx root*
 3. If $f(c) = 0$ return c *//Found it!*
 4. If $f(c) \cdot f(a) < 0$, $b \leftarrow c$
 5. If $f(c) \cdot f(b) < 0$, $a \leftarrow c$



The Bisection Method

THE BISECTION METHOD

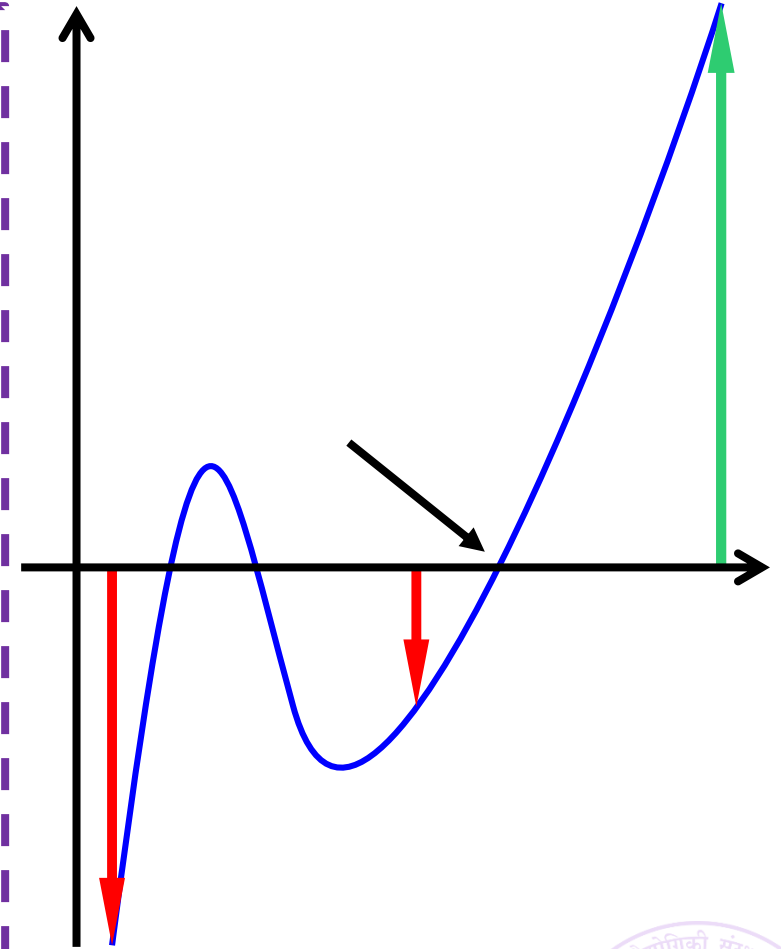
1. Given: continuous function f , active region $[a, b]$, tolerance $\epsilon > 0$
2. Check $f(a) \cdot f(b) < 0$ *//To apply IVT*
3. Repeat
 1. Let $c = \frac{1}{2}(a + b)$ *//Mid point*
 2. If $b - a < \epsilon$ return c *//Approx root*
 3. If $f(c) = 0$ return c *//Found it!*
 4. If $f(c) \cdot f(a) < 0$, $b \leftarrow c$
 5. If $f(c) \cdot f(b) < 0$, $a \leftarrow c$



The Bisection Method

THE BISECTION METHOD

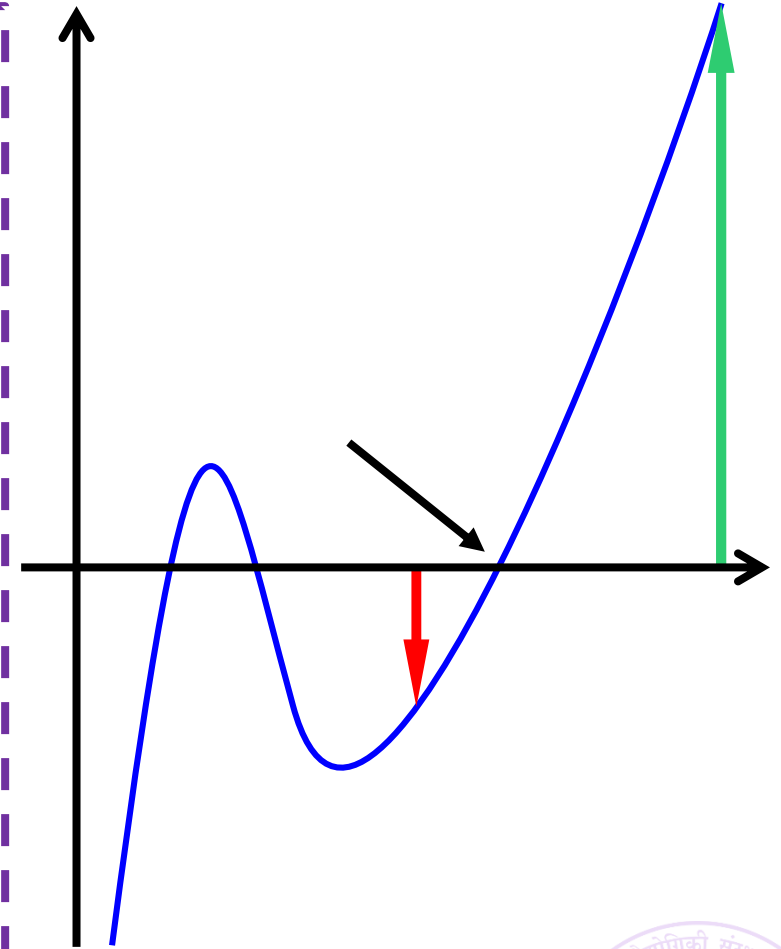
1. Given: continuous function f , active region $[a, b]$, tolerance $\epsilon > 0$
2. Check $f(a) \cdot f(b) < 0$ *//To apply IVT*
3. Repeat
 1. Let $c = \frac{1}{2}(a + b)$ *//Mid point*
 2. If $b - a < \epsilon$ return c *//Approx root*
 3. If $f(c) = 0$ return c *//Found it!*
 4. If $f(c) \cdot f(a) < 0$, $b \leftarrow c$
 5. If $f(c) \cdot f(b) < 0$, $a \leftarrow c$



The Bisection Method

THE BISECTION METHOD

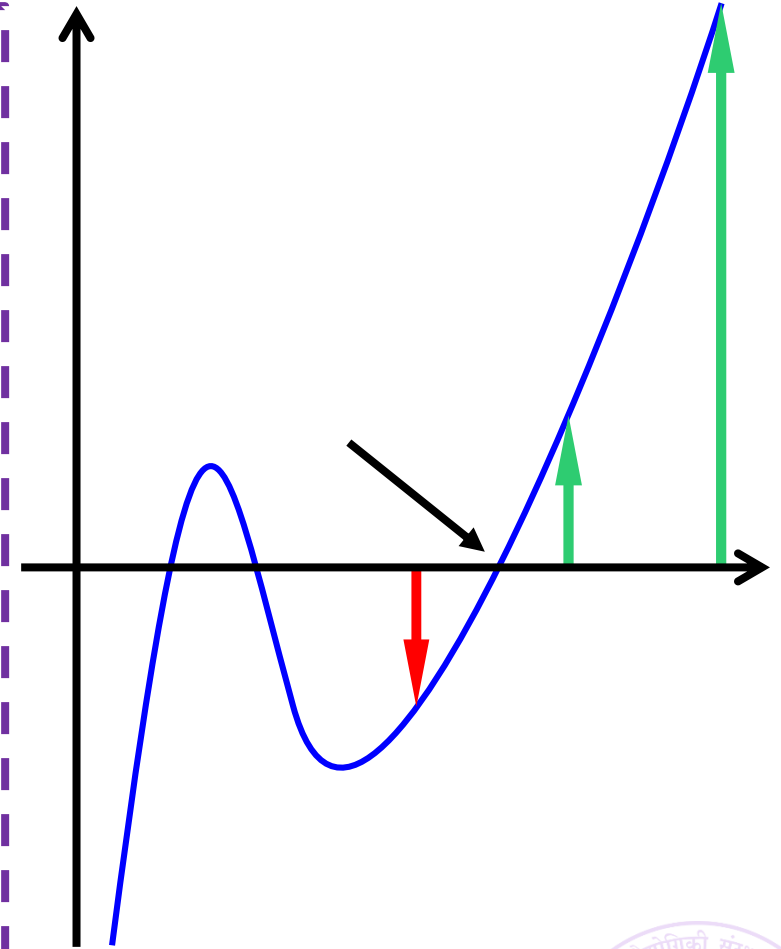
1. Given: continuous function f , active region $[a, b]$, tolerance $\epsilon > 0$
2. Check $f(a) \cdot f(b) < 0$ *//To apply IVT*
3. Repeat
 1. Let $c = \frac{1}{2}(a + b)$ *//Mid point*
 2. If $b - a < \epsilon$ return c *//Approx root*
 3. If $f(c) = 0$ return c *//Found it!*
 4. If $f(c) \cdot f(a) < 0$, $b \leftarrow c$
 5. If $f(c) \cdot f(b) < 0$, $a \leftarrow c$



The Bisection Method

THE BISECTION METHOD

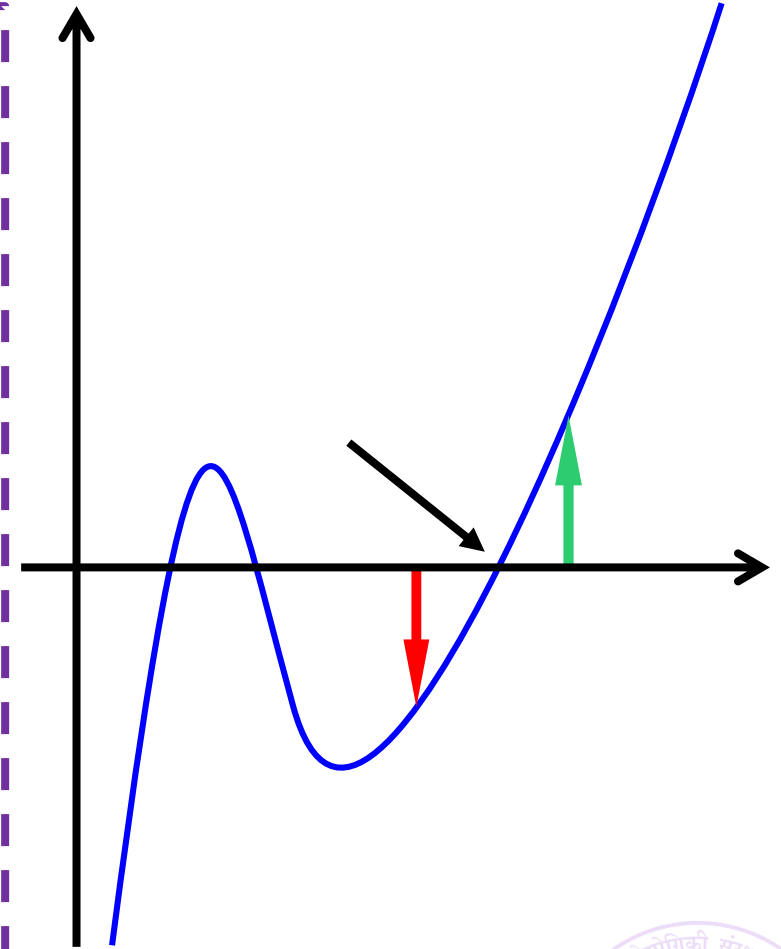
1. Given: continuous function f , active region $[a, b]$, tolerance $\epsilon > 0$
2. Check $f(a) \cdot f(b) < 0$ *//To apply IVT*
3. Repeat
 1. Let $c = \frac{1}{2}(a + b)$ *//Mid point*
 2. If $b - a < \epsilon$ return c *//Approx root*
 3. If $f(c) = 0$ return c *//Found it!*
 4. If $f(c) \cdot f(a) < 0$, $b \leftarrow c$
 5. If $f(c) \cdot f(b) < 0$, $a \leftarrow c$



The Bisection Method

THE BISECTION METHOD

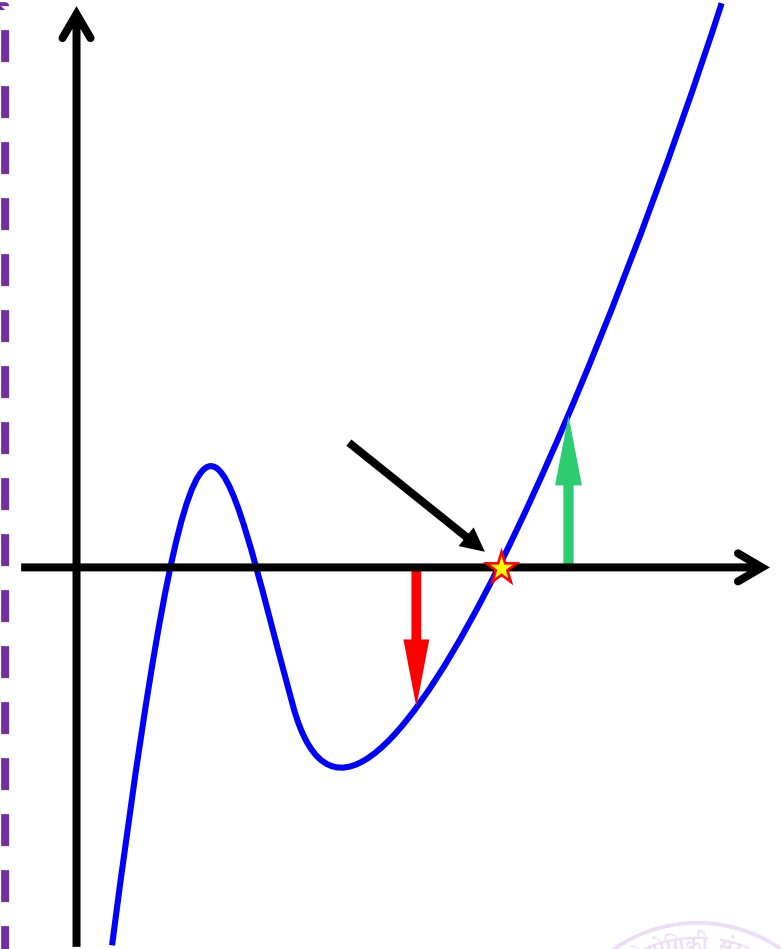
1. Given: continuous function f , active region $[a, b]$, tolerance $\epsilon > 0$
2. Check $f(a) \cdot f(b) < 0$ *//To apply IVT*
3. Repeat
 1. Let $c = \frac{1}{2}(a + b)$ *//Mid point*
 2. If $b - a < \epsilon$ return c *//Approx root*
 3. If $f(c) = 0$ return c *//Found it!*
 4. If $f(c) \cdot f(a) < 0$, $b \leftarrow c$
 5. If $f(c) \cdot f(b) < 0$, $a \leftarrow c$



The Bisection Method

THE BISECTION METHOD

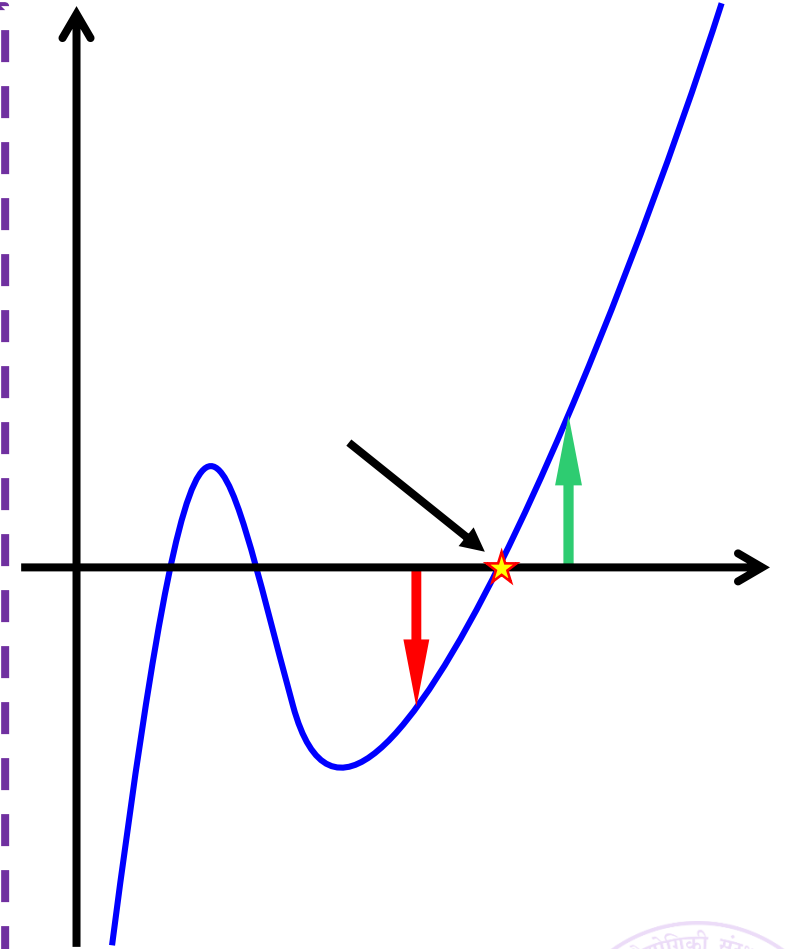
1. Given: continuous function f , active region $[a, b]$, tolerance $\epsilon > 0$
2. Check $f(a) \cdot f(b) < 0$ *//To apply IVT*
3. Repeat
 1. Let $c = \frac{1}{2}(a + b)$ *//Mid point*
 2. If $b - a < \epsilon$ return c *//Approx root*
 3. If $f(c) = 0$ return c *//Found it!*
 4. If $f(c) \cdot f(a) < 0$, $b \leftarrow c$
 5. If $f(c) \cdot f(b) < 0$, $a \leftarrow c$



The Bisection Method

THE BISECTION METHOD

1. Given: continuous function f , active region $[a, b]$, tolerance $\epsilon > 0$
2. Check $f(a) \cdot f(b) < 0$ *//To apply IVT*
3. Repeat
 1. Let $c = \frac{1}{2}(a + b)$ *//Mid point*
 2. If $b - a < \epsilon$ return c *//Approx root*
 3. If $f(c) = 0$ return c *//Found it!*
 4. If $f(c) \cdot f(a) < 0$, $b \leftarrow c$
 5. If $f(c) \cdot f(b) < 0$, $a \leftarrow c$

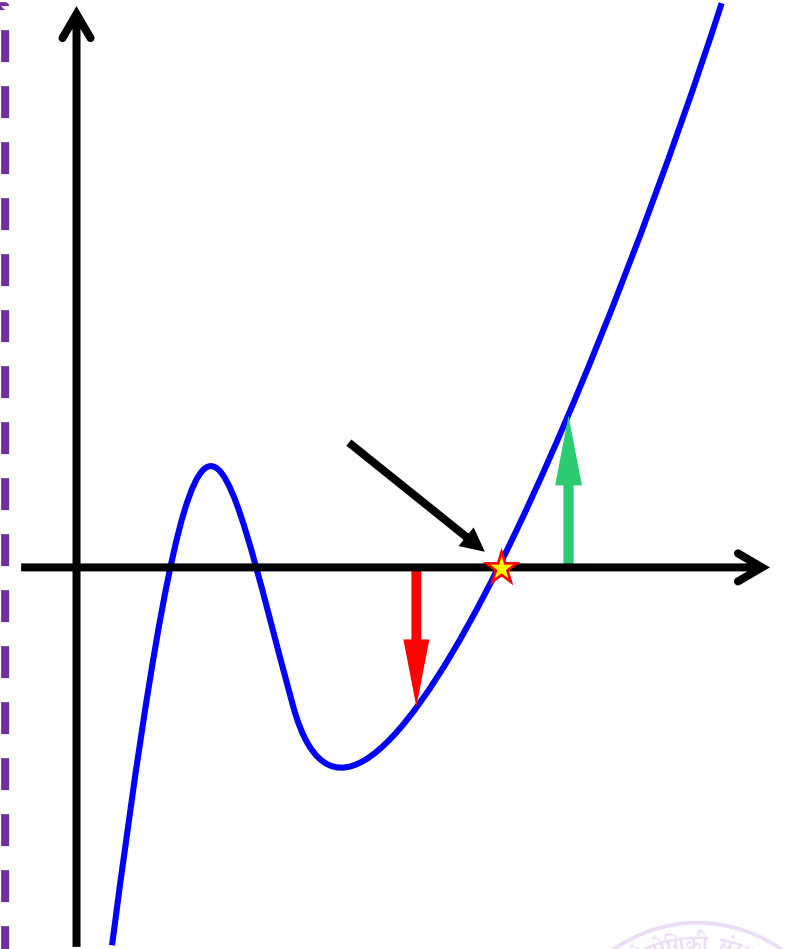


Can you show that this method stops in $\mathcal{O}\left(\log \frac{1}{\epsilon}\right)$ iterations?

The Bisection Method

THE BISECTION METHOD

1. Given: continuous function f , active region $[a, b]$, tolerance $\epsilon > 0$
2. Check $f(a) \cdot f(b) < 0$ *//To apply IVT*
3. Repeat
 1. Let $c = \frac{1}{2}(a + b)$ *//Mid point*
 2. If $b - a < \epsilon$ return c *//Approx root*
 3. If $f(c) = 0$ return c *//Found it!*
 4. If $f(c) \cdot f(a) < 0$, $b \leftarrow c$
 5. If $f(c) \cdot f(b) < 0$, $a \leftarrow c$



Can you show that this method stops in $\mathcal{O}\left(\log\frac{1}{\epsilon}\right)$ iterations?

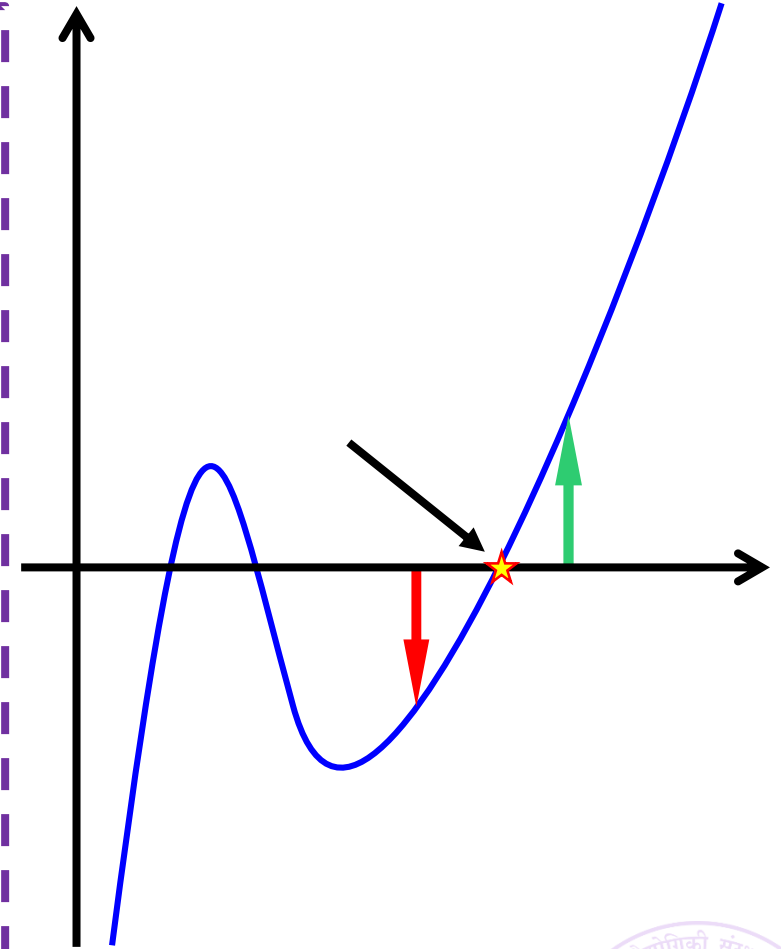
Can you find the square root of numbers using this method? Hint $f(x) = x^2 - v$

The Bisection Method

THE BISECTION METHOD

1. Given: continuous function f , active region $[a, b]$, tolerance $\epsilon > 0$
2. Check $f(a) \cdot f(b) < 0$ *//To apply IVT*
3.

Some of these intuitions extend to higher dimensions. If you are interested, check out www.youtube.com/watch?v=b7FxFsqfkOY
3. If $f(c) = 0$ return c *//Found it!*
4. If $f(c) \cdot f(a) < 0$, $b \leftarrow c$
5. If $f(c) \cdot f(b) < 0$, $a \leftarrow c$



Can you show that this method stops in $\mathcal{O}\left(\log\frac{1}{\epsilon}\right)$ iterations?

Can you find the square root of numbers using this method? Hint $f(x) = x^2 - v$

Finding Roots of Functions

The Newton Raphson Method



The Newton-Raphson Method

NEWTON-RAPHSON METHOD

1. Initialize x
2. Repeat
 1. Approximate f by $g(y) = f(x) + f'(x) \cdot (y - x)$
 2. Update $x \leftarrow \text{ROOT}(g) = x - \frac{f(x)}{f'(x)}$

The Newton-Raphson Method

NEWTON-RAPHSON METHOD

1. Initialize x
2. Repeat
 1. Approximate f by $g(y) = f(x) + f'(x) \cdot (y - x)$
 2. Update $x \leftarrow \text{ROOT}(g) = x - \frac{f(x)}{f'(x)}$

Tangent to f at x

The Newton-Raphson Method

NEWTON-RAPHSON METHOD

1. Initialize x

2. Repeat

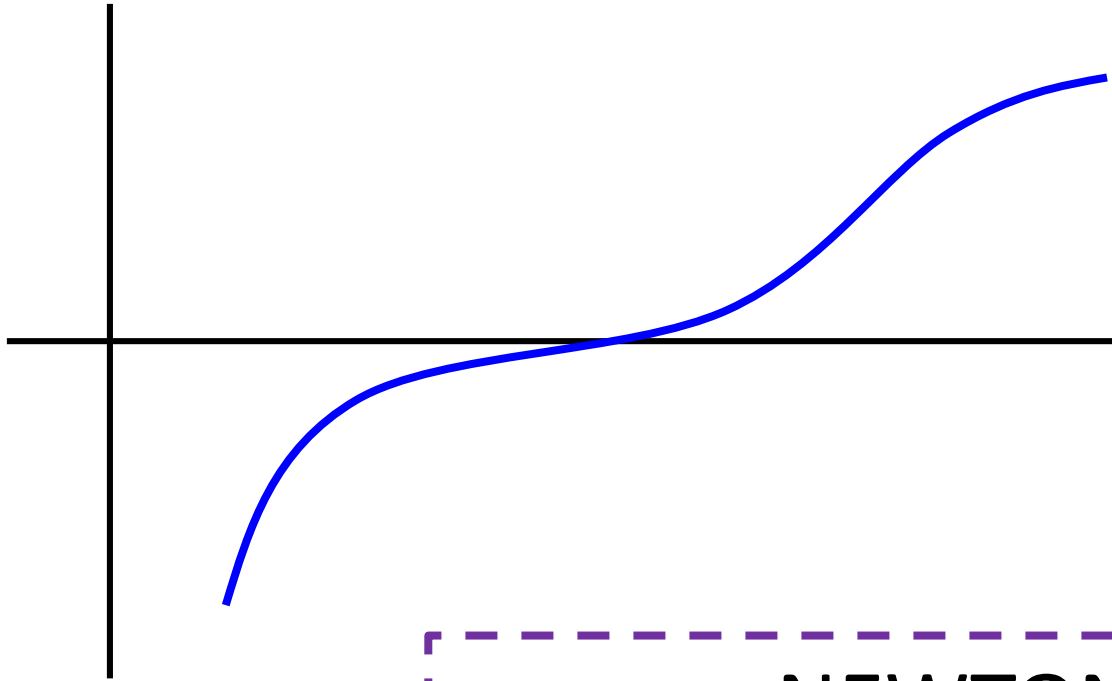
1. Approximate f by $g(y) = f(x) + f'(x)(y - x)$

2. Update $x \leftarrow \text{ROOT}(g) = x - \frac{f(x)}{f'(x)}$

Tangent to f at x

Where tangent intersects x axis

The Newton-Raphson Method



NEWTON-RAPHSON METHOD

1. Initialize x

2. Repeat

1. Approximate f by $g(y) = f(x) + f'(x)(y - x)$

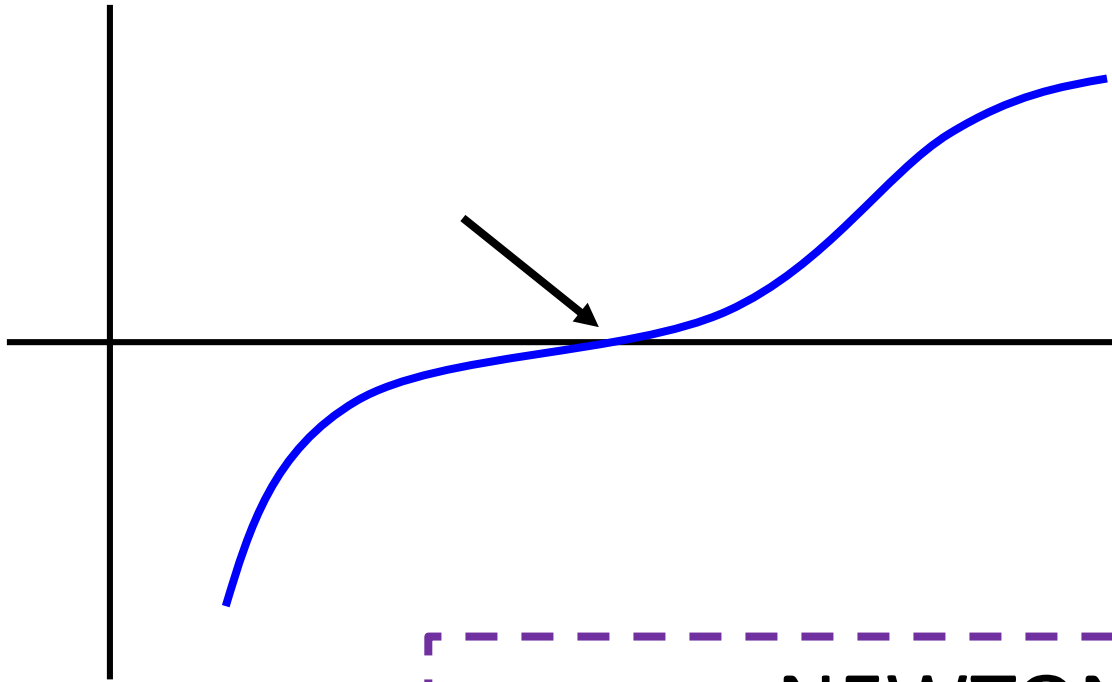
2. Update $x \leftarrow \text{ROOT}(g) = x - \frac{f(x)}{f'(x)}$

Tangent to f at x

Where tangent intersects x axis

The Newton-Raphson Method

$$x : f(x) = 0$$



NEWTON-RAPHSON METHOD

1. Initialize x

2. Repeat

1. Approximate f by $g(y) = f(x) + f'(x)(y - x)$

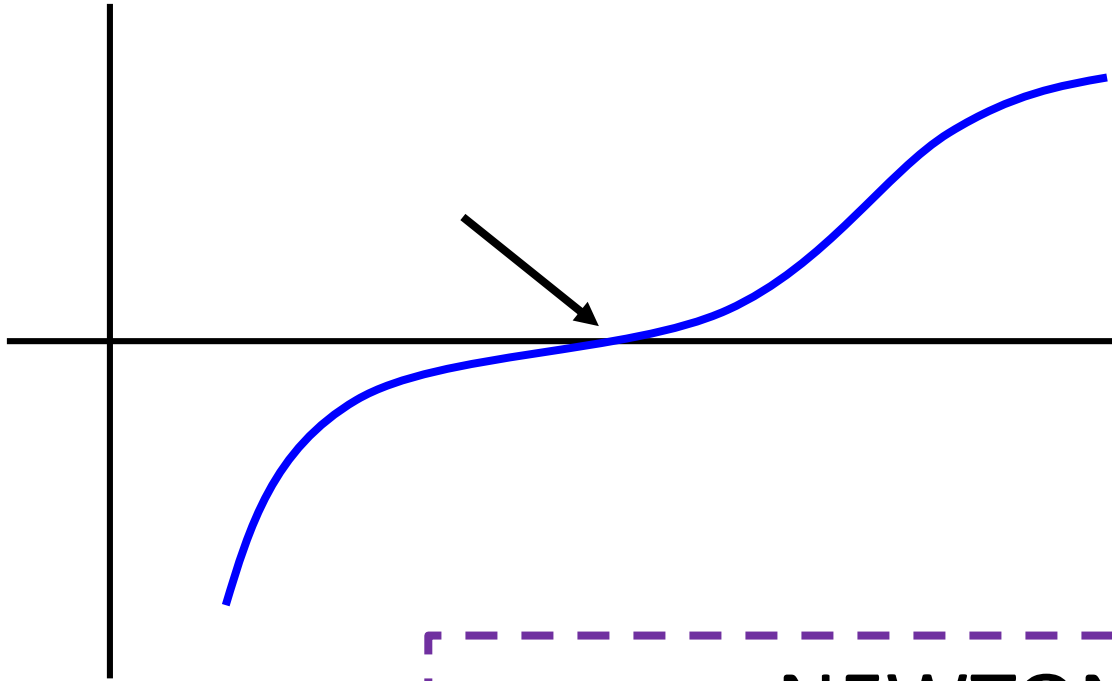
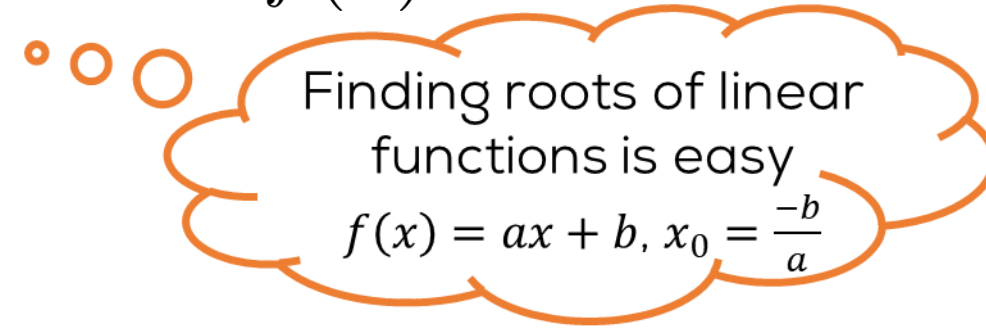
2. Update $x \leftarrow \text{ROOT}(g) = x - \frac{f(x)}{f'(x)}$

Tangent to f at x

Where tangent intersects x axis

The Newton-Raphson Method

$$x : f(x) = 0$$



NEWTON-RAPHSON METHOD

1. Initialize x

2. Repeat

1. Approximate f by $g(y) = f(x) + f'(x)(y - x)$

2. Update $x \leftarrow \text{ROOT}(g) = x - \frac{f(x)}{f'(x)}$

Tangent to f at x

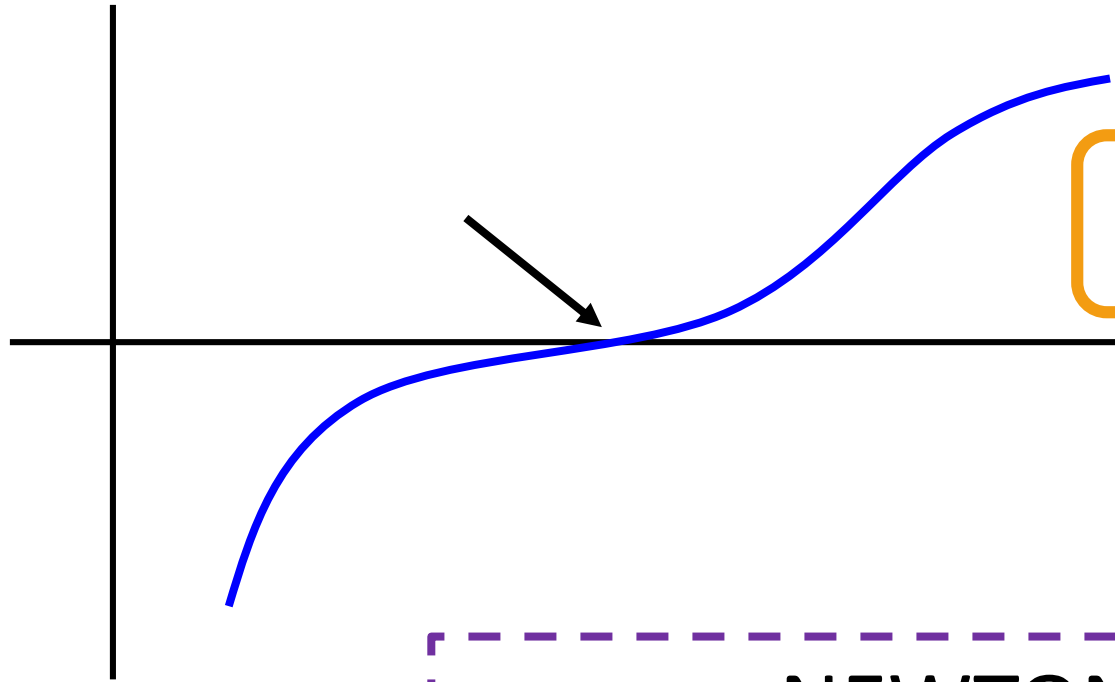
Where tangent intersects x axis

The Newton-Raphson Method

$$x : f(x) = 0$$

Finding roots of linear functions is easy
 $f(x) = ax + b, x_0 = \frac{-b}{a}$

But my function is nonlinear ☹



NEWTON-RAPHSON METHOD

1. Initialize x
2. Repeat

Tangent to f at x

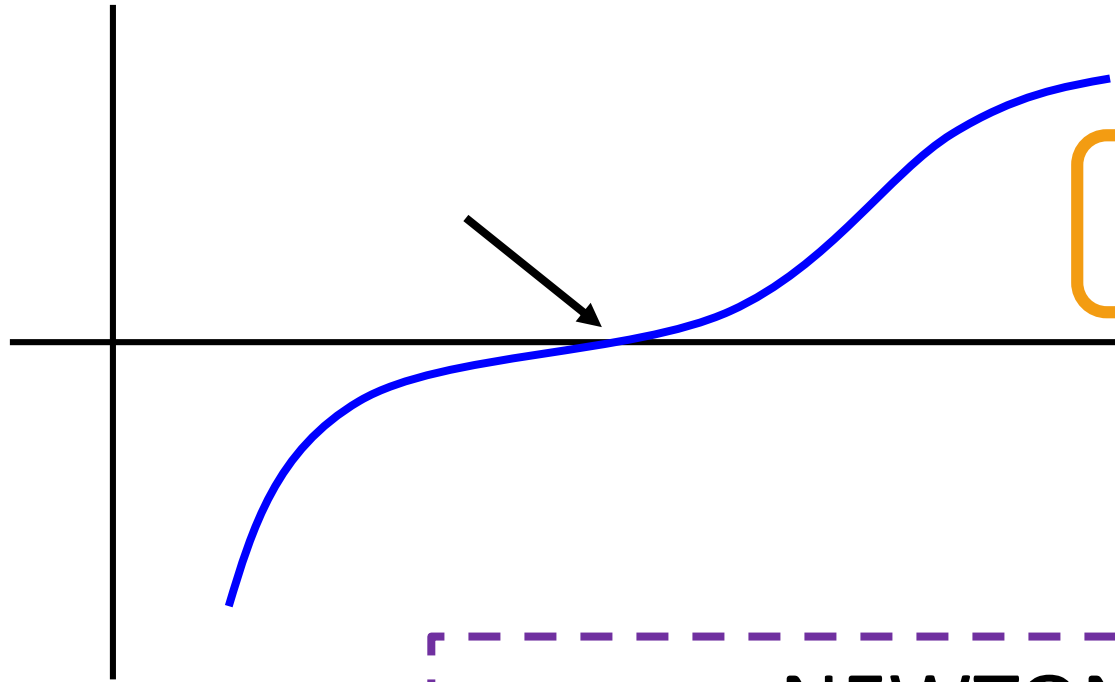
1. Approximate f by $g(y) = f(x) + f'(x)(y - x)$

2. Update $x \leftarrow \text{ROOT}(g) = x - \frac{f(x)}{f'(x)}$

Where tangent intersects x axis

The Newton-Raphson Method

$$x : f(x) = 0$$



Finding roots of linear functions is easy

$$f(x) = ax + b, x_0 = \frac{-b}{a}$$

But my function is nonlinear ☹️

Take a linear approximation to it

NEWTON-RAPHSON METHOD

1. Initialize x

2. Repeat

1. Approximate f by $g(y) = f(x) + f'(x)(y - x)$

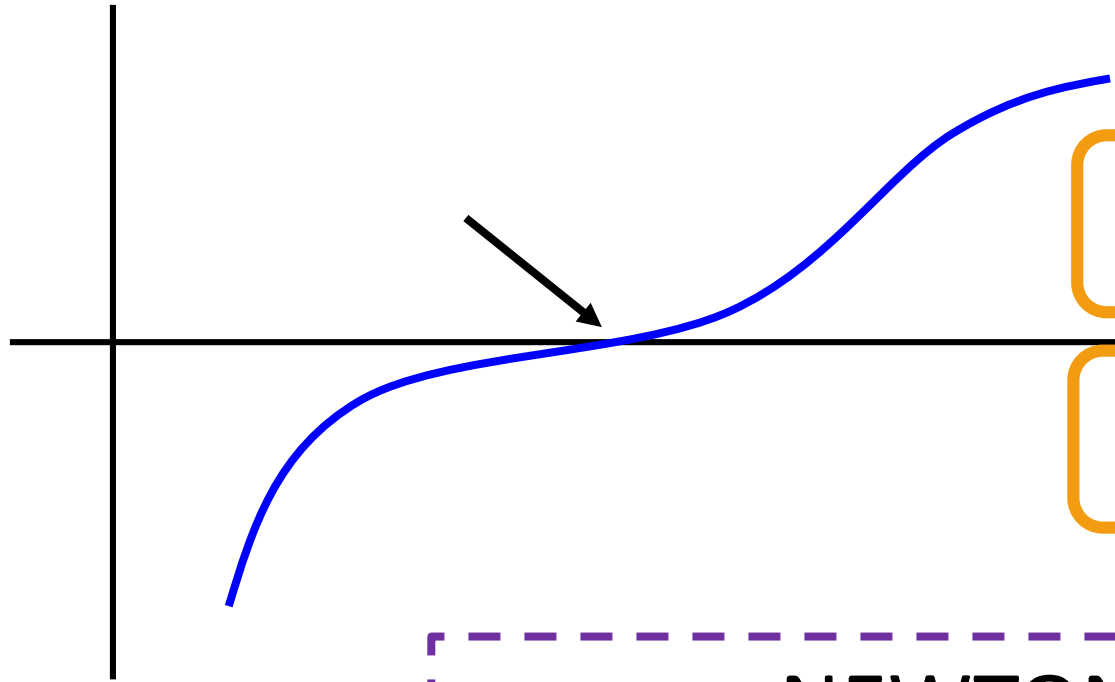
2. Update $x \leftarrow \text{ROOT}(g) = x - \frac{f(x)}{f'(x)}$

Tangent to f at x

Where tangent intersects x axis

The Newton-Raphson Method

$$x : f(x) = 0$$



Finding roots of linear functions is easy

$$f(x) = ax + b, x_0 = \frac{-b}{a}$$

But my function is nonlinear ☹️

Take a linear approximation to it

How does one do that?

NEWTON-RAPHSON METHOD

1. Initialize x

2. Repeat

1. Approximate f by $g(y) = f(x) + f'(x)(y - x)$

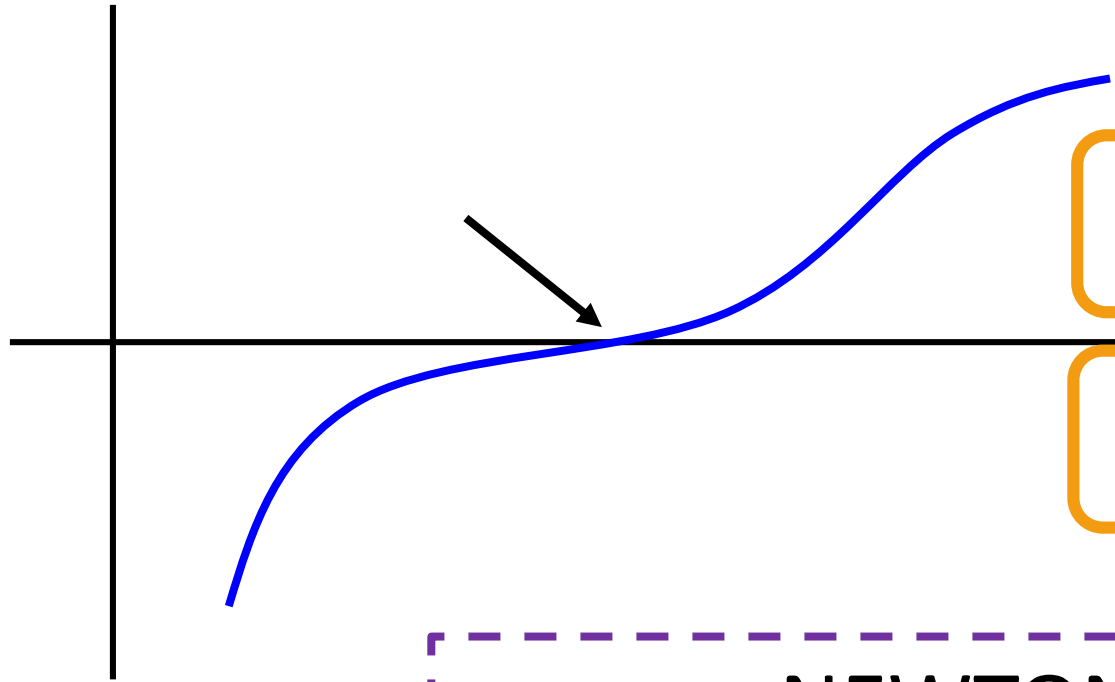
2. Update $x \leftarrow \text{ROOT}(g) = x - \frac{f(x)}{f'(x)}$

Tangent to f at x

Where tangent intersects x axis

The Newton-Raphson Method

$$x : f(x) = 0$$



Finding roots of linear functions is easy

$$f(x) = ax + b, x_0 = \frac{-b}{a}$$

But my function is nonlinear ☹️

How does one do that?

Take a linear approximation to it

Many ways e.g. Taylor's expansion

NEWTON-RAPHSON METHOD

1. Initialize x

2. Repeat

1. Approximate f by $g(y) = f(x) + f'(x)(y - x)$

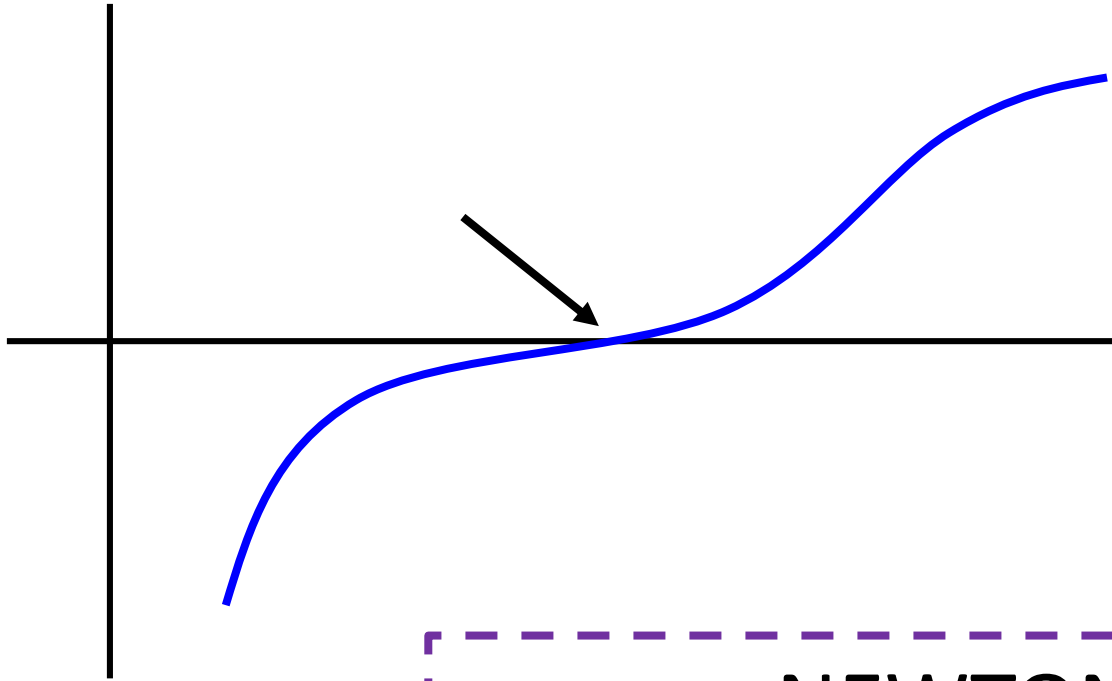
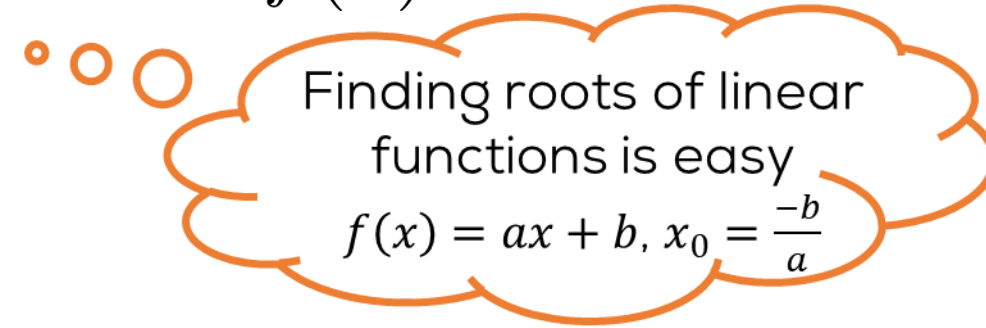
2. Update $x \leftarrow \text{ROOT}(g) = x - \frac{f(x)}{f'(x)}$

Tangent to f at x

Where tangent intersects x axis

The Newton-Raphson Method

$$x : f(x) = 0$$



NEWTON-RAPHSON METHOD

1. Initialize x

2. Repeat

1. Approximate f by $g(y) = f(x) + f'(x)(y - x)$

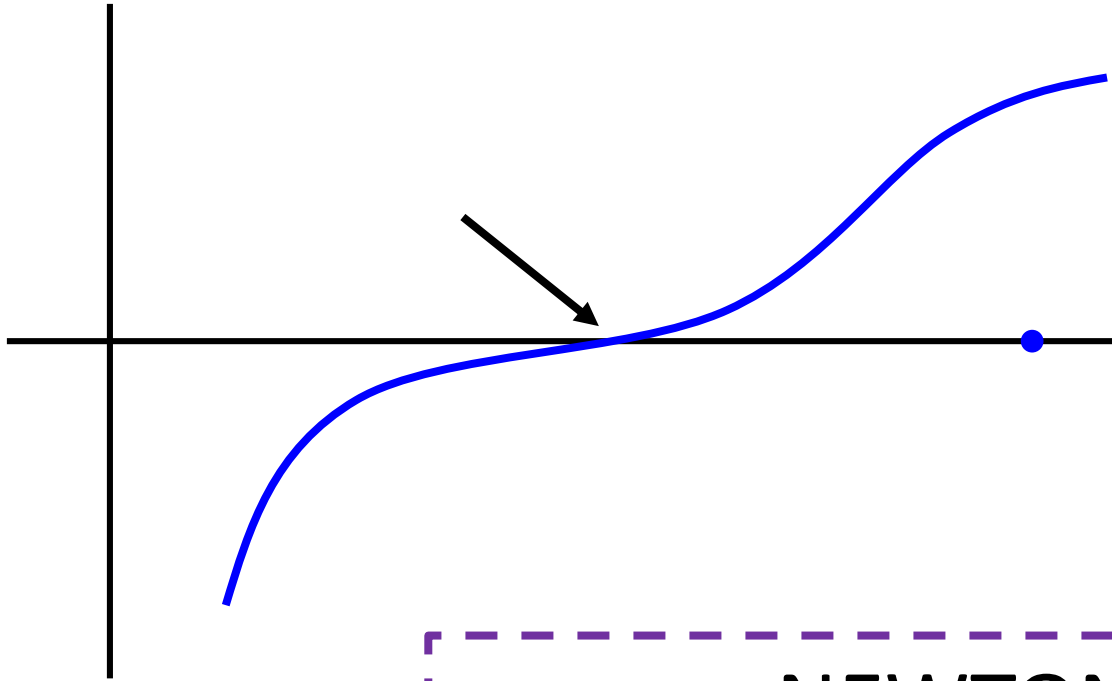
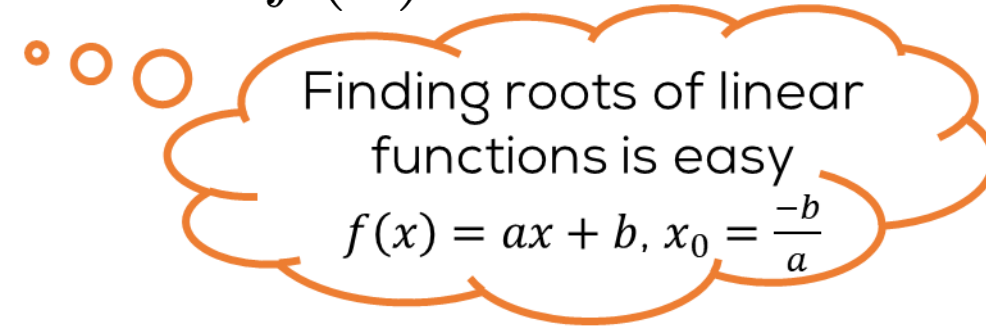
2. Update $x \leftarrow \text{ROOT}(g) = x - \frac{f(x)}{f'(x)}$

Tangent to f at x

Where tangent intersects x axis

The Newton-Raphson Method

$$x : f(x) = 0$$



NEWTON-RAPHSON METHOD

1. Initialize x

2. Repeat

1. Approximate f by $g(y) = f(x) + f'(x)(y - x)$

2. Update $x \leftarrow \text{ROOT}(g) = x - \frac{f(x)}{f'(x)}$

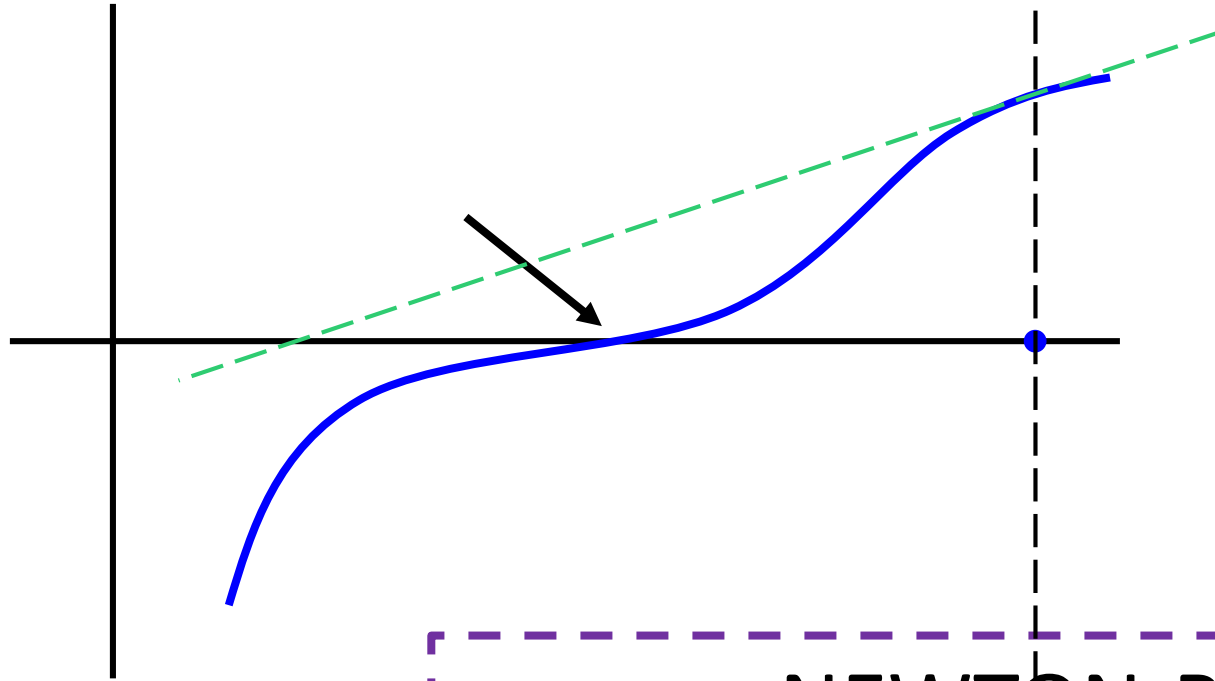
Tangent to f at x

Where tangent intersects x axis

The Newton-Raphson Method

$$x : f(x) = 0$$

Finding roots of linear functions is easy
 $f(x) = ax + b, x_0 = \frac{-b}{a}$



NEWTON-RAPHSON METHOD

1. Initialize x

2. Repeat

1. Approximate f by $g(y) = f(x) + f'(x)(y - x)$

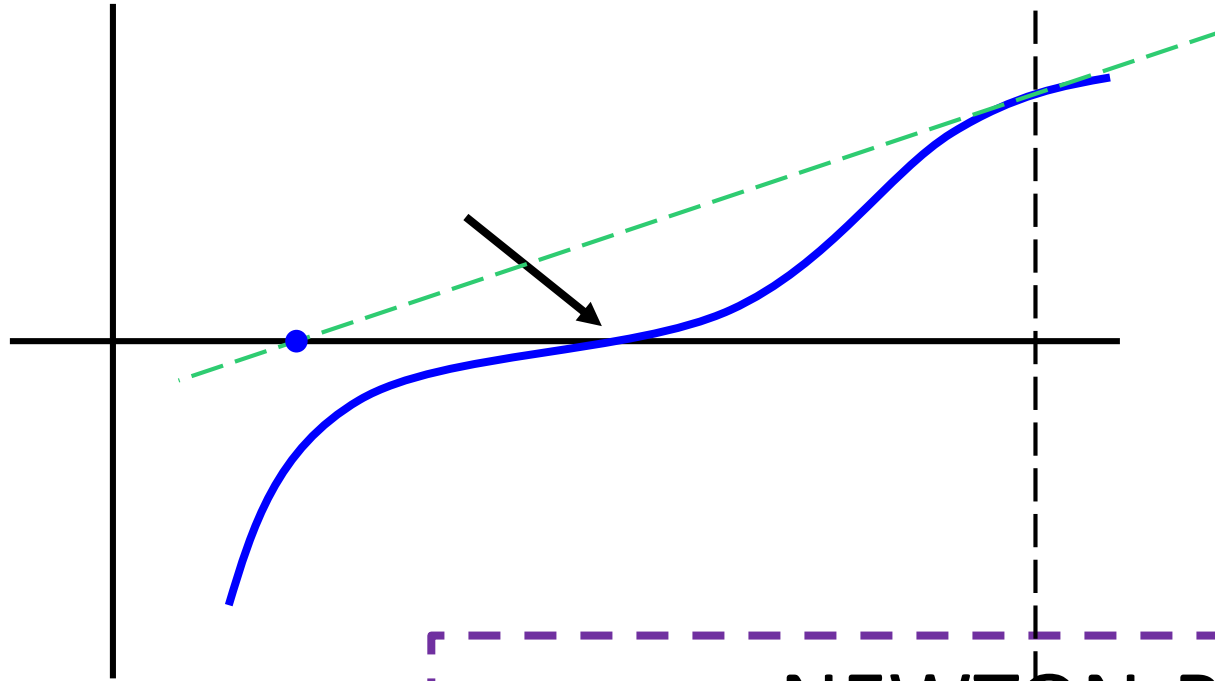
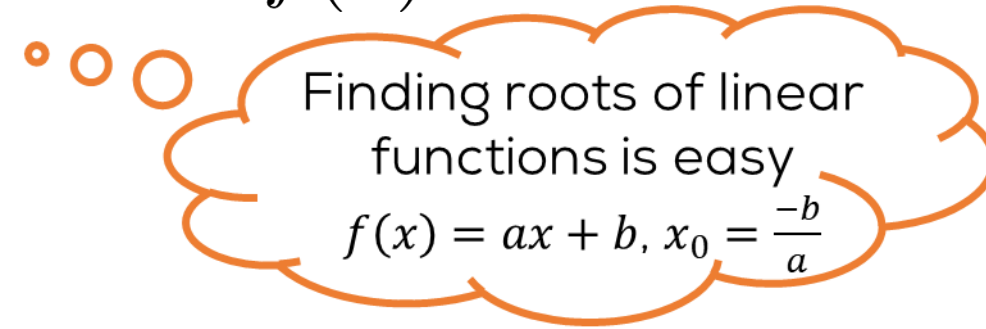
2. Update $x \leftarrow \text{ROOT}(g) = x - \frac{f(x)}{f'(x)}$

Tangent to f at x

Where tangent intersects x axis

The Newton-Raphson Method

$$x : f(x) = 0$$



NEWTON-RAPHSON METHOD

1. Initialize x

2. Repeat

1. Approximate f by $g(y) = f(x) + f'(x)(y - x)$

2. Update $x \leftarrow \text{ROOT}(g) = x - \frac{f(x)}{f'(x)}$

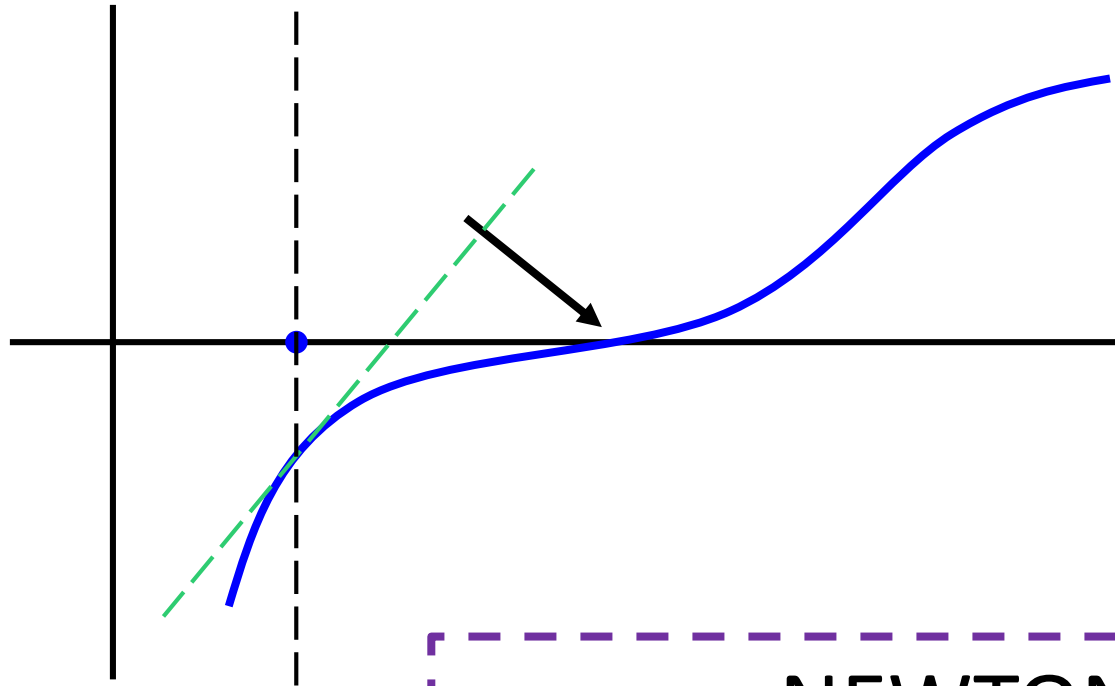
Tangent to f at x

Where tangent intersects x axis

The Newton-Raphson Method

$$x : f(x) = 0$$

Finding roots of linear functions is easy
 $f(x) = ax + b, x_0 = \frac{-b}{a}$



NEWTON-RAPHSON METHOD

1. Initialize x

2. Repeat

1. Approximate f by $g(y) = f(x) + f'(x)(y - x)$

2. Update $x \leftarrow \text{ROOT}(g) = x - \frac{f(x)}{f'(x)}$

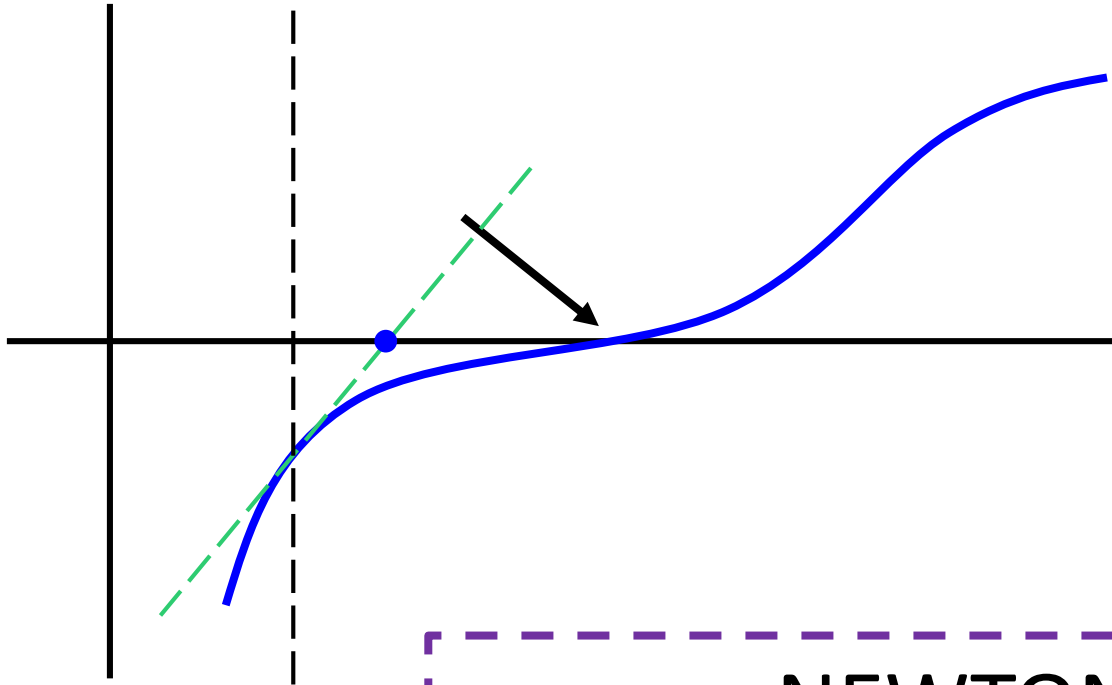
Tangent to f at x

Where tangent intersects x axis

The Newton-Raphson Method

$$x : f(x) = 0$$

Finding roots of linear functions is easy
 $f(x) = ax + b, x_0 = \frac{-b}{a}$



NEWTON-RAPHSON METHOD

1. Initialize x

2. Repeat

1. Approximate f by $g(y) = f(x) + f'(x)(y - x)$

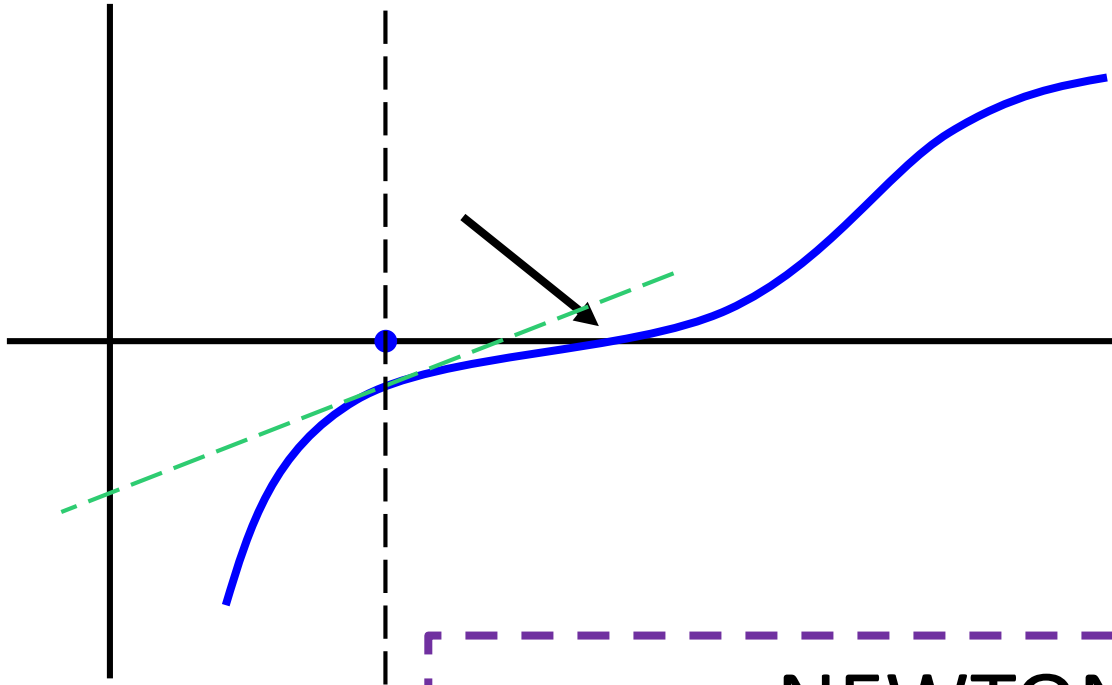
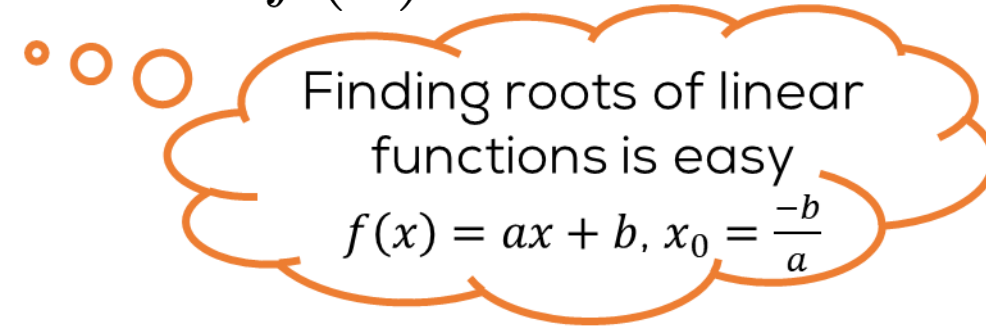
2. Update $x \leftarrow \text{ROOT}(g) = x - \frac{f(x)}{f'(x)}$

Tangent to f at x

Where tangent intersects x axis

The Newton-Raphson Method

$$x : f(x) = 0$$



NEWTON-RAPHSON METHOD

1. Initialize x

2. Repeat

1. Approximate f by $g(y) = f(x) + f'(x)(y - x)$

2. Update $x \leftarrow \text{ROOT}(g) = x - \frac{f(x)}{f'(x)}$

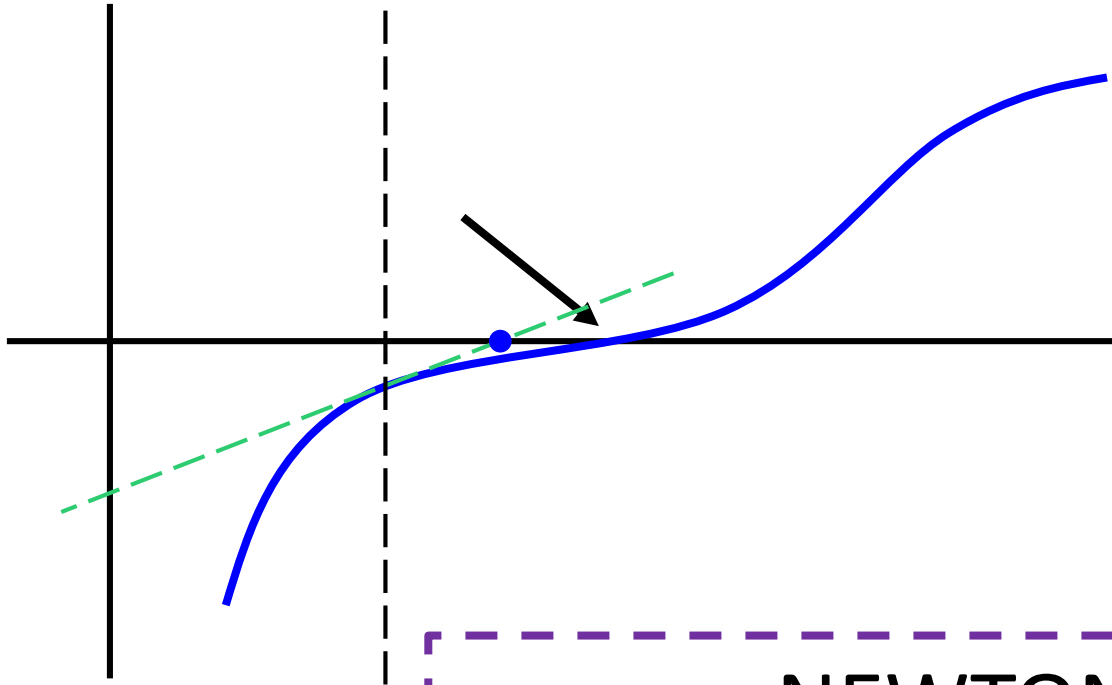
Tangent to f at x

Where tangent intersects x axis

The Newton-Raphson Method

$$x : f(x) = 0$$

Finding roots of linear functions is easy
 $f(x) = ax + b, x_0 = \frac{-b}{a}$



NEWTON-RAPHSON METHOD

1. Initialize x

2. Repeat

1. Approximate f by $g(y) = f(x) + f'(x)(y - x)$

2. Update $x \leftarrow \text{ROOT}(g) = x - \frac{f(x)}{f'(x)}$

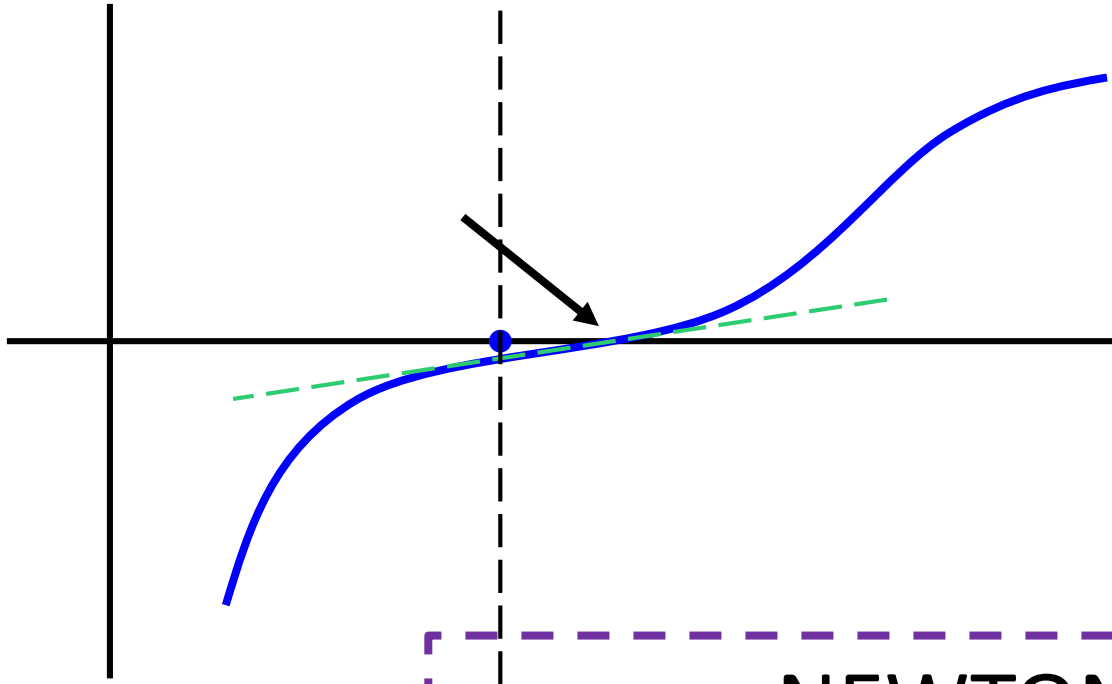
Tangent to f at x

Where tangent intersects x axis

The Newton-Raphson Method

$$x : f(x) = 0$$

Finding roots of linear functions is easy
 $f(x) = ax + b, x_0 = \frac{-b}{a}$



NEWTON-RAPHSON METHOD

1. Initialize x

2. Repeat

1. Approximate f by $g(y) = f(x) + f'(x)(y - x)$

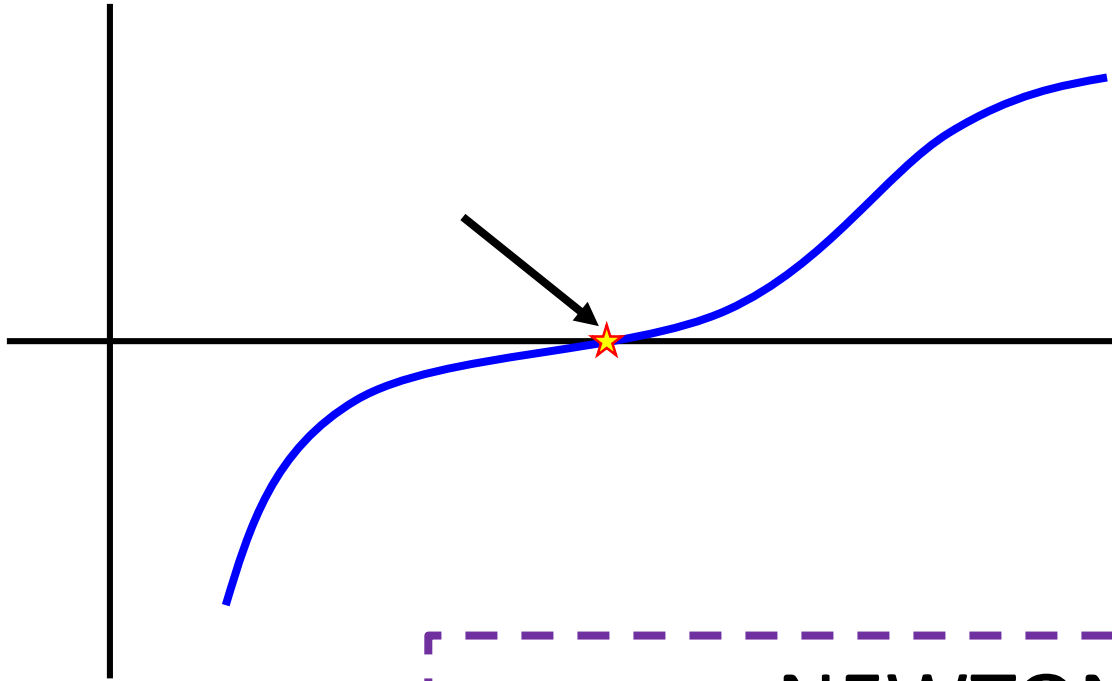
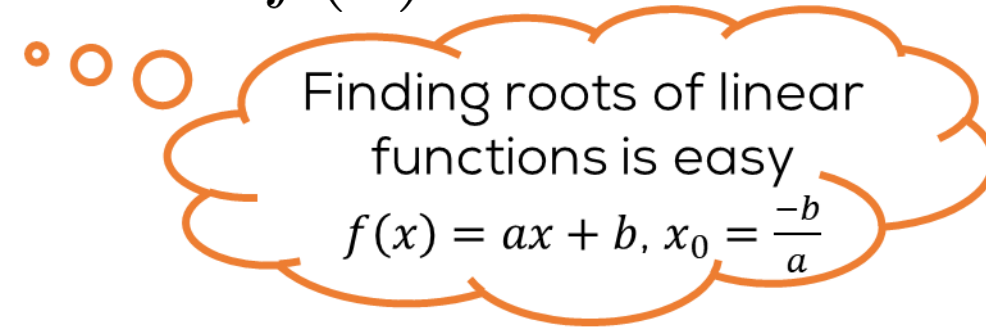
2. Update $x \leftarrow \text{ROOT}(g) = x - \frac{f(x)}{f'(x)}$

Tangent to f at x

Where tangent intersects x axis

The Newton-Raphson Method

$$x : f(x) = 0$$



NEWTON-RAPHSON METHOD

1. Initialize x

2. Repeat

1. Approximate f by $g(y) = f(x) + f'(x)(y - x)$

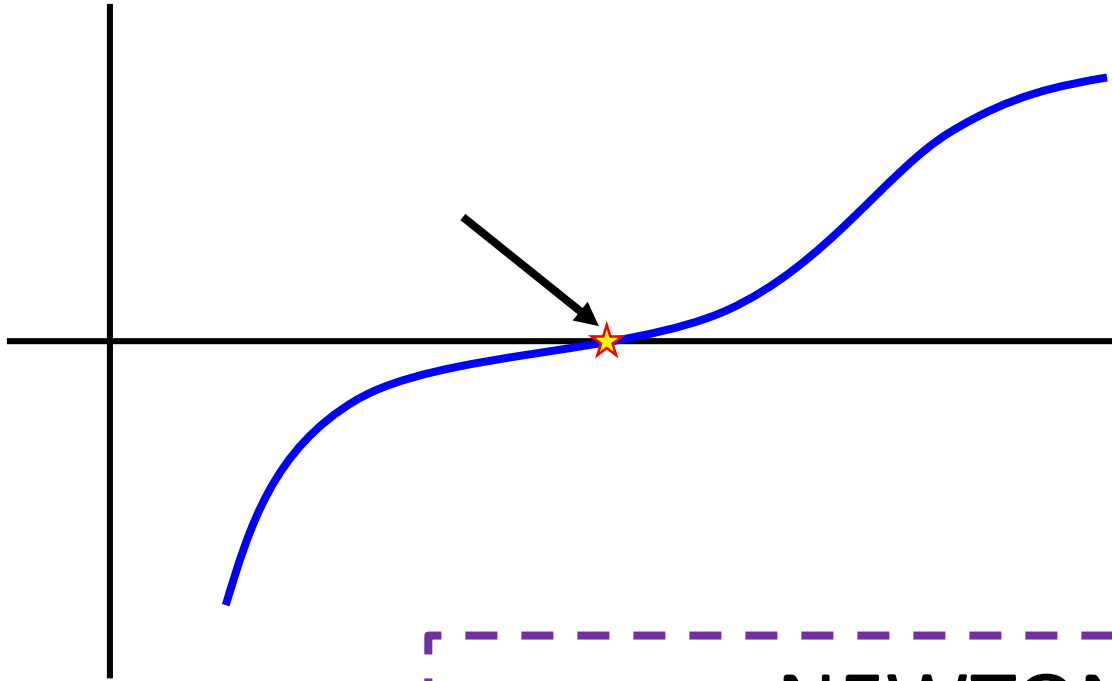
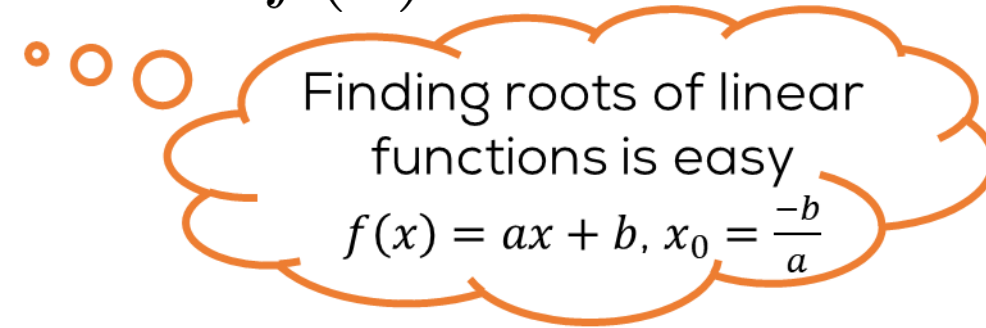
2. Update $x \leftarrow \text{ROOT}(g) = x - \frac{f(x)}{f'(x)}$

Tangent to f at x

Where tangent intersects x axis

The Newton-Raphson Method

$$x : f(x) = 0$$



No guarantees in general. Depends a lot on initialization

NEWTON-RAPHSON METHOD

1. Initialize x

2. Repeat

Tangent to f at x

1. Approximate f by $g(y) = f(x) + f'(x)(y - x)$

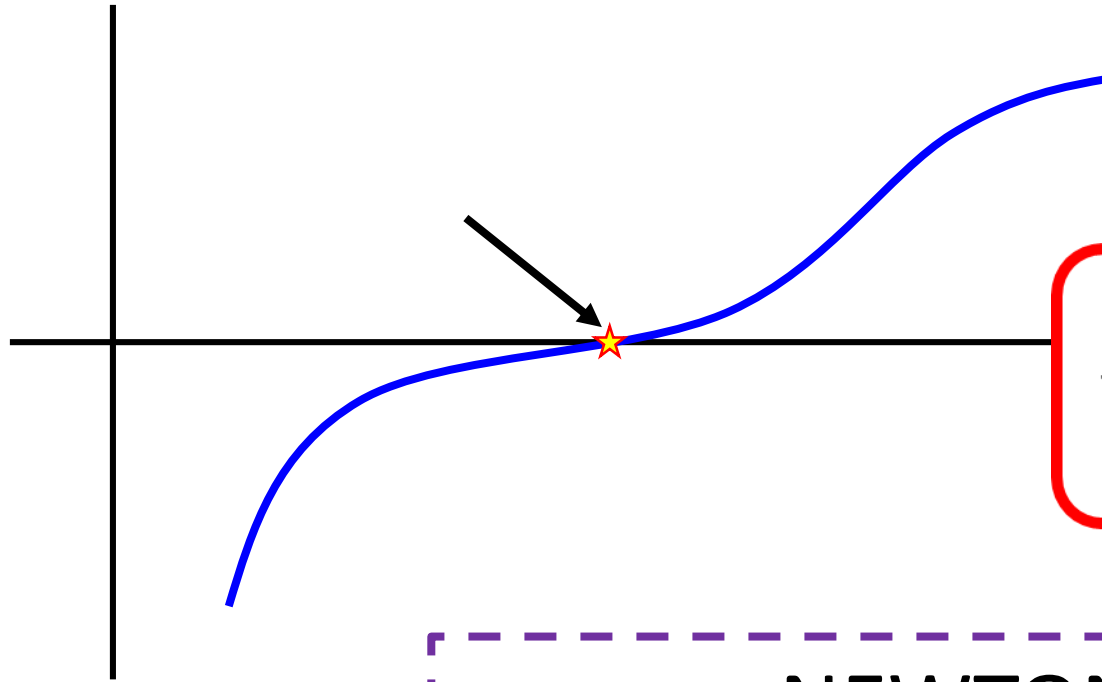
2. Update $x \leftarrow \text{ROOT}(g) = x - \frac{f(x)}{f'(x)}$

Where tangent intersects x axis

The Newton-Raphson Method

$$x : f(x) = 0$$

Finding roots of linear functions is easy
 $f(x) = ax + b, x_0 = \frac{-b}{a}$



What does NR look like if used to find square root of numbers $f(x) = x^2 - v$? Pleasant surprise awaits 😊

No guarantees in general. Depends a lot on initialization

NEWTON-RAPHSON METHOD

1. Initialize x
2. Repeat

Tangent to f at x

1. Approximate f by $g(y) = f(x) + f'(x)(y - x)$

2. Update $x \leftarrow \text{ROOT}(g) = x - \frac{f(x)}{f'(x)}$

Where tangent intersects x axis

The Newton-Raphson Method

$$x : f(x) = 0$$

Finding roots of linear functions is easy

$$f(x) = ax + b, x_0 = \frac{-b}{a}$$

Also check out the secant method and the Regula Falsi method (do not require derivative calculation)

What does NR look like if used to find square root of numbers $f(x) = x^2 - v$? Pleasant surprise awaits 😊

No guarantees in general. Depends a lot on initialization

NEWTON-RAPHSON METHOD

1. Initialize x
2. Repeat

Tangent to f at x

1. Approximate f by $g(y) = f(x) + f'(x)(y - x)$

2. Update $x \leftarrow \text{ROOT}(g) = x - \frac{f(x)}{f'(x)}$

Where tangent intersects x axis