Please Grab a handout in the back.

Today

Solving Nonlinear Function

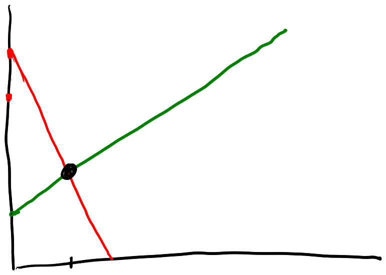
Bracketing Methods: Most important: Bisection

Announcements / Reminders

Lab 1 due Thursday - Check Auto-grader!

HW 1 Released (Graded)

Linear

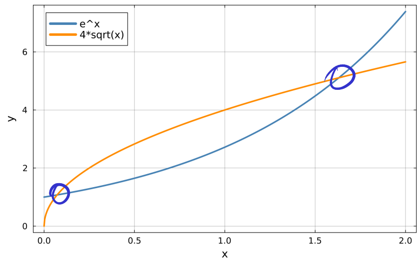


$-2x + 4 = x + 1$

$x_2 = -2x_1 + 4$ $\quad$ $x_2 = x_1 + 1$

$\boxed{x = 1}$

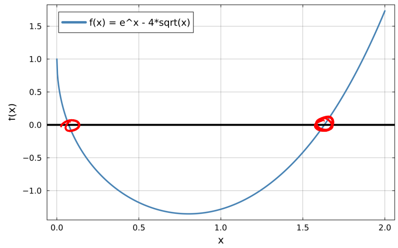
Nonlinear



$e^x = 4\sqrt{x}$

no analytical in "simple functions"

$\Rightarrow$



$f(x) = 0 = e^x - 4\sqrt{x}$

find "roots" x values where $f(x) = 0$

Bracketing $\qquad$ Open

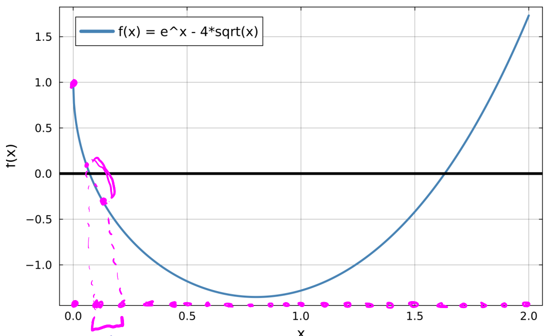
Today

Incremental Search

Incremental Search

input : function f, search range $[x_{\min}, x_{\max}]$, number of steps n
output: list of brackets $[x_l, x_u]$, each containing a root

$\Delta x \leftarrow (x_{\max} - x_{\min}) / n$
brackets $\leftarrow \{\}$
$x_l \leftarrow x_{\min}$
for $k \leftarrow 1$ **to** n **do**
 $x_u \leftarrow x_l + \Delta x$
 if $f(x_l) \cdot f(x_u) < 0$ **then**
  // sign change $\Rightarrow$ a root lies in $[x_l, x_u]$
  append $[x_l, x_u]$ to brackets
 end
 $x_l \leftarrow x_u$
end
return brackets



if $f(x)$ is continuous and $f(x_l)$ and $f(x_r)$ have different signs, a root is between x_l and x_r

Strength: Can find multiple roots

Weakness: Can miss roots, slow at finding one root (we will see shortly)

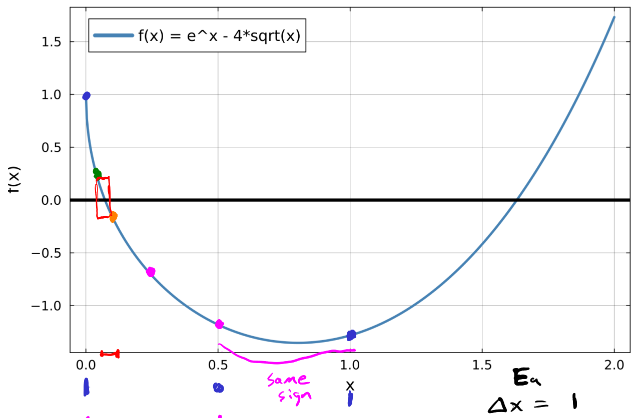
error $\rightarrow E_a = \frac{\Delta x}{n-1}$

Bisection

Bisection

input : function f, bracket $[x_l, x_u]$ with $f(x_l) \cdot f(x_u) < 0$,
 tolerance ε_s
output: estimate x_r of the root

$x_r \leftarrow x_l$
repeat
 $x_{r,\text{old}} \leftarrow x_r$
 $x_r \leftarrow (x_l + x_u) / 2$
 // keep the half that still brackets the root
 if $f(x_l) \cdot f(x_r) < 0$ **then**
  $x_u \leftarrow x_r$
 else if $f(x_l) \cdot f(x_r) > 0$ **then**
  $x_l \leftarrow x_r$
 else
  return x_r $\qquad$ // exact root
 end
 $\varepsilon_a \leftarrow \left| \frac{x_r - x_{r,\text{old}}}{x_r} \right| \cdot 100\%$
until $\varepsilon_a < \varepsilon_s$
return x_r



Iter: 1, 2, 3, 4, 5

0.0675 $\quad$ 0.125

E_a

$\Delta x = 1$

$\frac{\Delta x}{2} = 0.5$

$\frac{\Delta x}{4} = 0.25$

$\frac{\Delta x}{8} = 0.125$

$\frac{\Delta x}{16} = 0.0675$

$\boxed{E_a = \frac{\Delta x}{2^n}}$

$$E_a = \frac{\Delta x}{2^n}$$

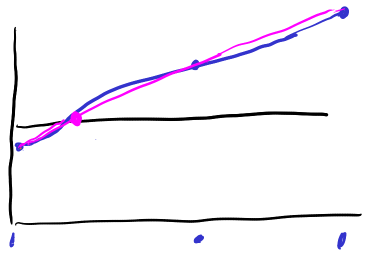
	Incremental	Bisection
function calls for desired $\frac{\Delta x}{E_s}$	$E_a = E_s = \frac{\Delta x}{n-1}$	$E_a = E_s = \frac{\Delta x}{2^n}$
	$n-1 = \frac{\Delta x}{E_s}$	$2^n = \frac{\Delta x}{E_s}$
	$\boxed{n = \frac{\Delta x}{E_s} + 1}$	$\boxed{n = \log_2\left(\frac{\Delta x}{E_s}\right)}$
	$O\left(\frac{\Delta x}{E_s}\right)$	$O\left(\log\left(\frac{\Delta x}{E_s}\right)\right)$
$E_s = 0.01\,\Delta x$, $\frac{\Delta x}{E_s} = 100$	101 func evals	$\log(100) = 6.64$; 7 function evals !
$E_s = 0.0001\,\Delta x$, $\frac{\Delta x}{E_s} = 10{,}000$	10,001 func evals	14 func evals

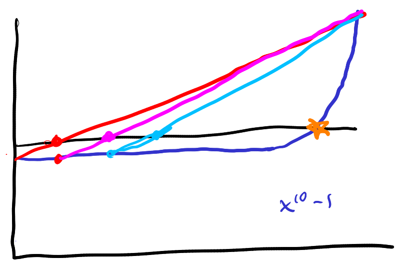
False Position

False Position

input : function f, bracket $[x_l, x_u]$ with $f(x_l) \cdot f(x_u) < 0$,
tolerance ε_s
output: estimate x_r of the root

$x_r \leftarrow x_l$
repeat
  $x_{r,\text{old}} \leftarrow x_r$
  $x_r \leftarrow x_u - \frac{f(x_u)\,(x_l - x_u)}{f(x_l) - f(x_u)}$ ⇐ only difference from bisection
  `// keep the side that still brackets the root`
  if $f(x_l) \cdot f(x_r) < 0$ **then**
    $x_u \leftarrow x_r$
  else if $f(x_l) \cdot f(x_r) > 0$ **then**
    $x_l \leftarrow x_r$
  else
    return x_r `// exact root`
  end
  $\varepsilon_a \leftarrow \left|\frac{x_r - x_{r,\text{old}}}{x_r}\right| \cdot 100\%$
until $\varepsilon_a < \varepsilon_s$
return x_r





Problem: Doesn't handle curvature well