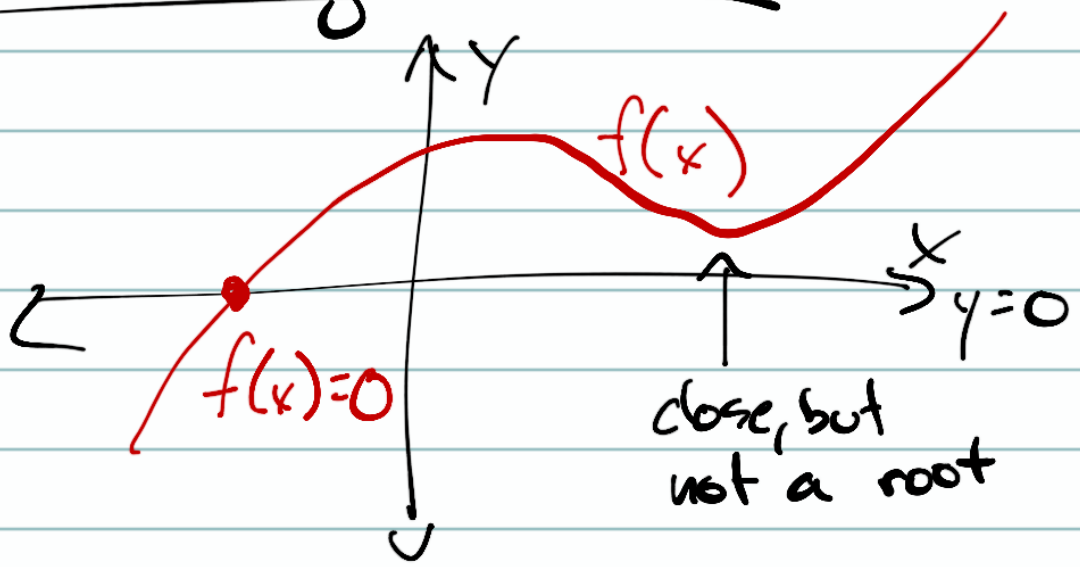
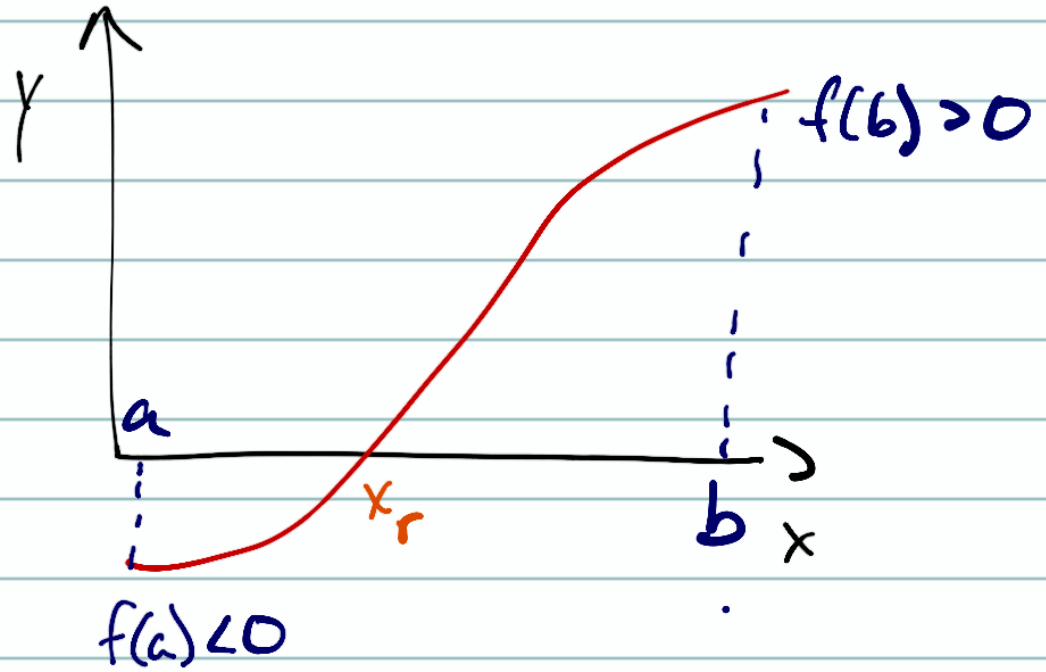
Root-Finding methods



Roots are values of x for which a function is zero (or equivalently where is equals a constant)

→ Most obvious method: Brute force - randomly choose values of x until a root, x_r, is found
- → Even looking within some interval $[a, b]$, there are an infinite number of points
- → Even when $f(x_r)$ is close to zero, this method does not guarantee to find a root here

Bisection Method



→ If we select a, b such that $f(a) \cdot f(b) < 0$, and $f(x)$ is continuous in this interval, then we are guaranteed to find a root

Guess $\boxed{x_1 = \frac{a+b}{2}}$ halfway point

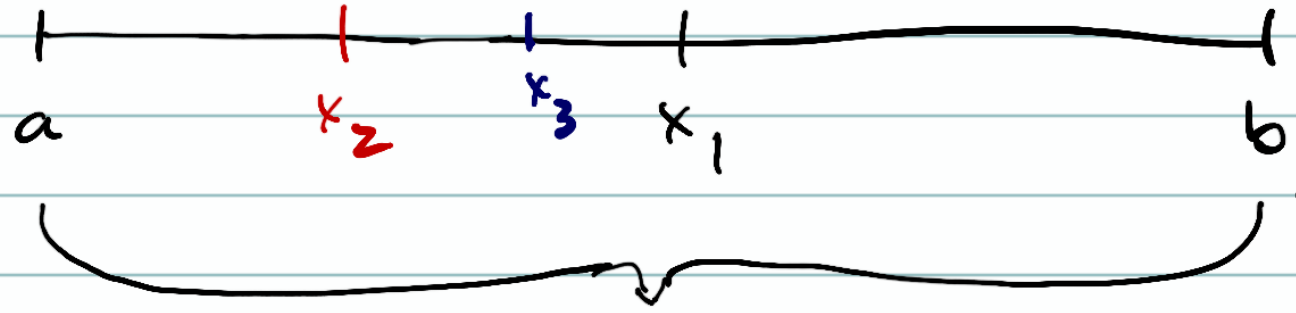
→ If $f(a) \cdot f(x_1) < 0$ and $f(x_1) \cdot f(b) > 0$
→ Then, replace b with x_1 and do again

→ If $f(a) \cdot f(x_1) > 0$ and $f(x_1) \cdot f(b) < 0$
→ Then, replace a with x_1 and do again

→ Error = b-a on each iteration

How fast do we approach the exact root?

(This is called the "convergence rate")



Initial range $\frac{b-a}{2^0}$

Range 1: $\frac{b-a}{2^1}$

Range 2: $\frac{b-a}{2^2}$

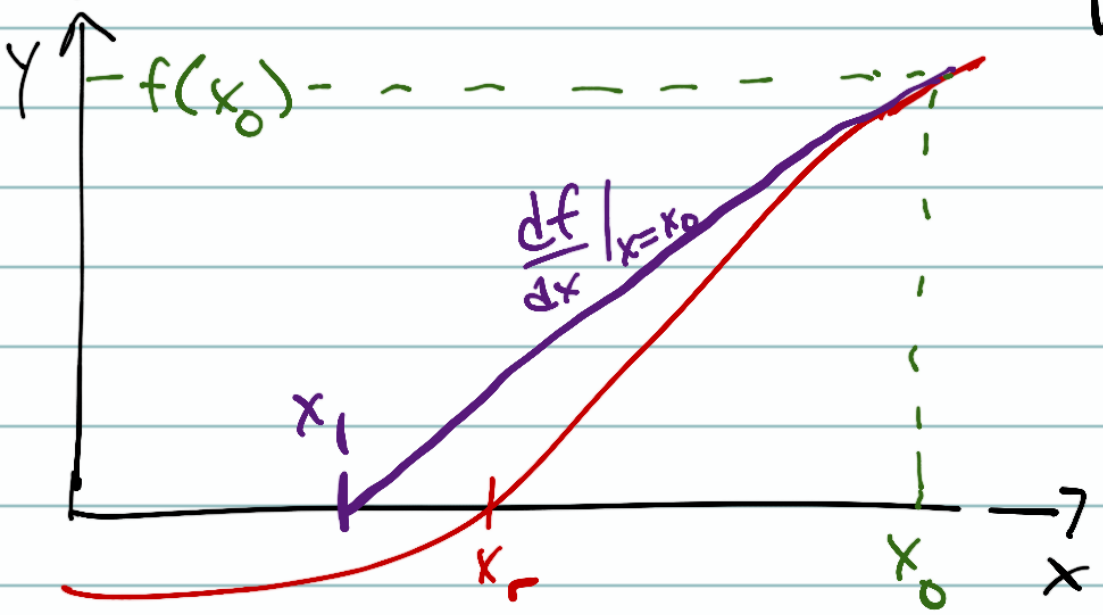
Range 3: $\frac{b-a}{2^3}$

In general max error at each iteration is:

$$|\varepsilon_n| < \frac{b-a}{2^n}$$

Newton-Raphson method

Reminder: The derivative is a handy way to tell us where $f(x)$ is going



-> Initial guess x_0

-> Notice $f(x_0) > 0$ and $\frac{df}{dx}|_{x=x_0} > 0$

-> Must move to left to find root

-> Make a tangent line and follow it to x-axis ($y=0$) to make next guess

Tangent line → $\frac{y-f(x_0)}{x-x_0} = \left.\frac{df}{dx}\right|_{x=x_0}$

Intersects $y=0$ → $0 = f(x_0) + \left.\frac{df}{dx}\right|_{x=x_0}(x_1 - x_0)$

$$x_1 = x_0 - \frac{f(x_0)}{\left.\frac{df}{dx}\right|_{x=x_0}}$$

Do this iteratively with

$$x_{n+1} = x_n - f(x_n) \Big/ \left.\frac{df}{dx}\right|_{x=x_n}$$

Newton-Raphson method

Algorithm (Hint: A while loop is useful for ②)

Guess x_0 (make sure $\left.\frac{df}{dx}\right|_{x=x_0} \neq 0$)

① Calculate $x_{n+1} = x_n - f(x_n) \Big/ \left.\frac{df}{dx}\right|_{x=x_0}$

② Test if $|x_{n+1} - x_n| <$ error threshold
If so → stop
If not → ①

Convergence of Newton-Raphson method is faster than bisection (though it does have issues with functions with $\frac{df}{dx}=0$ in places)

What if we have a system of equations with many variables, that we want to find the root of:

$$f_1(x_1,\dots,x_n)=0$$
$$f_2(x_1,\dots,x_n)=0$$
$$\vdots$$
$$f_n(x_1,\dots,x_n)=0$$

We can construct a matrix version of Newton-Raphson (iterations k)

$$\begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}_{k+1} = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}_{k} - \begin{bmatrix} & F & \end{bmatrix} \begin{bmatrix} f_1(\vec{x}_k) \\ \vdots \\ f_n(\vec{x}_k) \end{bmatrix}$$

Where $F = (J^T J)^{-1} J^T$

with ^{T} indicating a matrix transpose
and $^{-1}$ indicating a matrix inverse

More on these next week.

- In the mean time
MATLAB → ' → matrix transposition
inv(A) → matrix inverse

J → jacobian matrix →

$$J = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & & \\ \frac{\partial f_n}{\partial x_1} & \cdots & \frac{\partial f_n}{\partial x_n} \end{bmatrix}$$

The derivs of all the functions with respect to all the variables

Otherwise the algorithm is the same.