Non-linear Equations

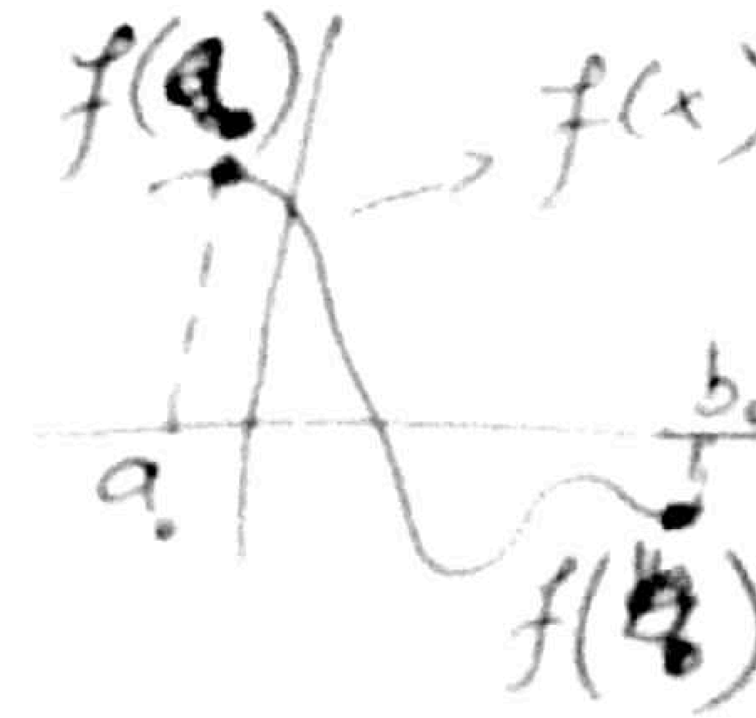
1. Root Finding:

Let $g(x) = h(x)$ be given [Find values of x when g and h are equal]

define $f(x) = g(x) - h(x)$

Target: Find roots of $f(x)$

Method 1: Bisection Method



Take Two points a_0 and b_0, such that

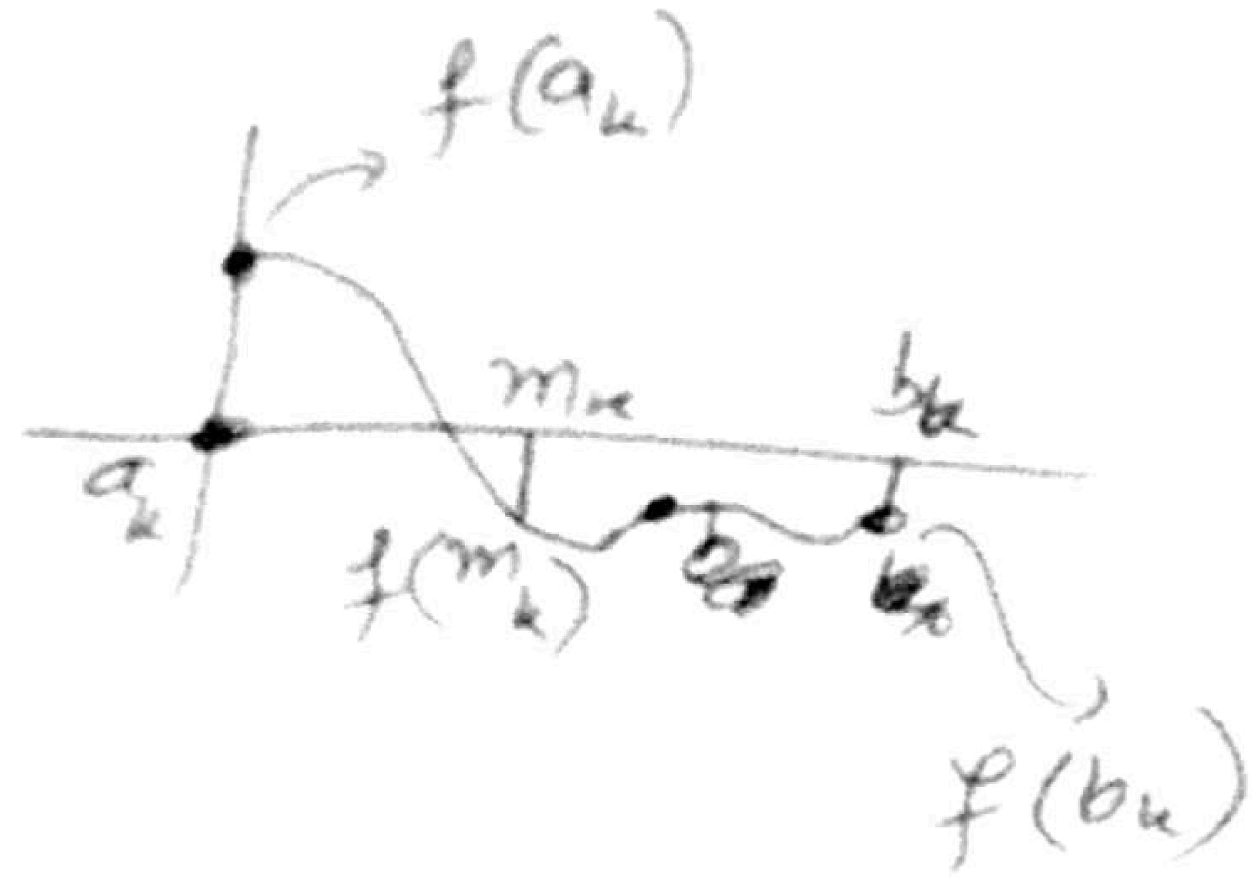
$f(a_0) > 0$, and $f(b_0) < 0$

So there is a point $\underbrace{\text{b/w } a_0 \text{ and } b_0}_{\to [a_0, b_0]}$ where f crosses the horizontal axis

Let us consider $m_k = \frac{a_k + b_k}{2}$ [at stage k]

1. If $f(m_k) < 0 \Rightarrow$

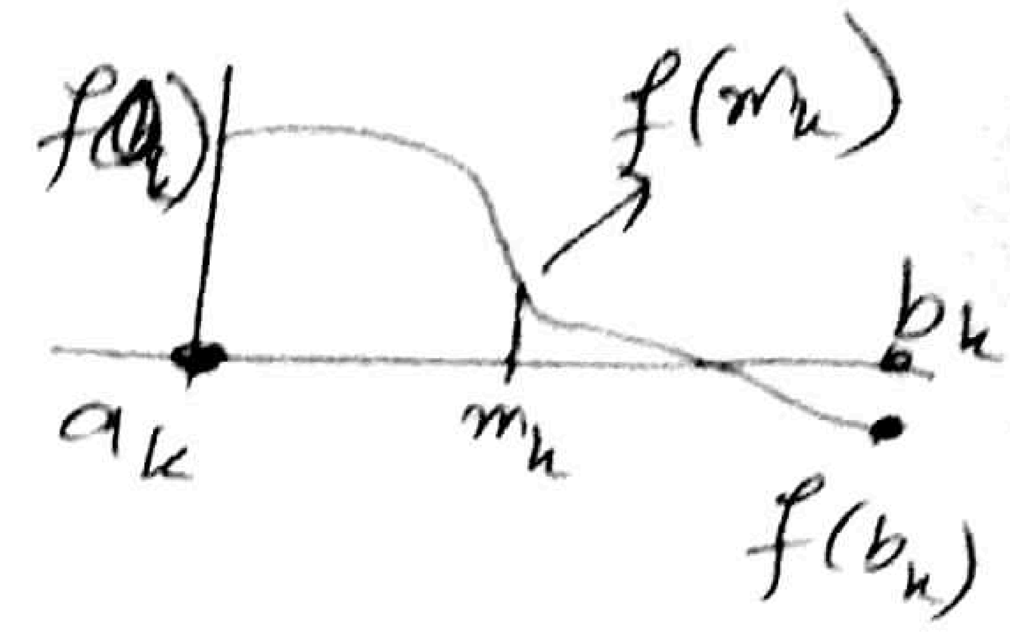
$\Rightarrow$ root is in $[a_k, m_k]$



So $a_{k+1} = a_k$

$b_{k+1} = m_k$

2. If $f(m_k) > 0$



$\Rightarrow$ Root is in $[m_k, b_k]$

so $a_{k+1} = m_k$, $b_{k+1} = b_k$

3. If $f(m_k) = 0$

Then m_k is the root. STOP.

(This rarely happens).

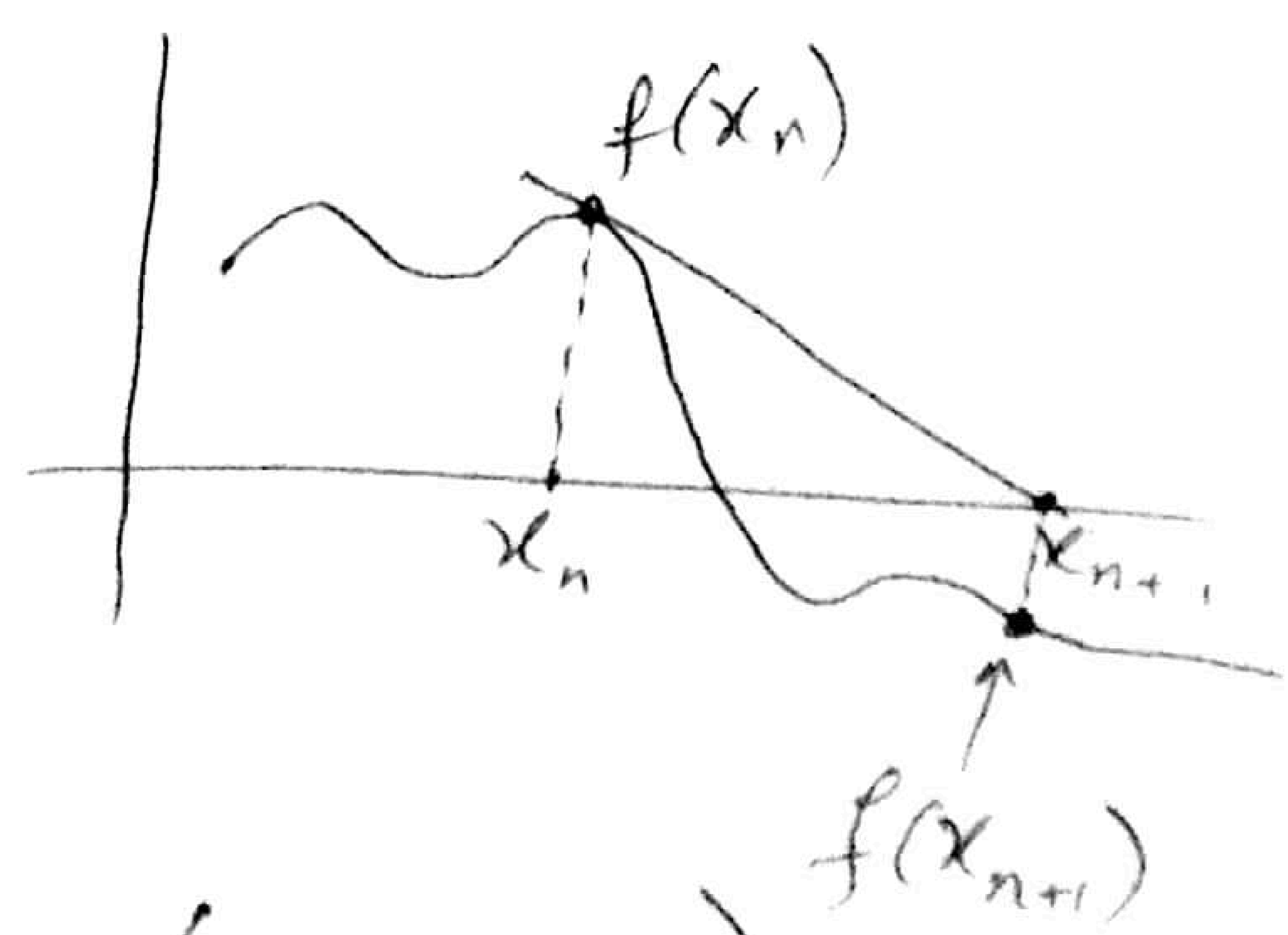
So, stopping criteria is $|b_k - a_k|$ is very small

like $|b_k - a_k| = 2^{-k} |b_0 - a_0|$.

- Algo is guaranteed to converge.
- Algo working time may be large.
- If there are many roots, it will give only one of them.

Method 2: Newton-Raphson:

To make faster, we look for tangent crossing the horizontal axis



As tangent is a line, so eq of line is

$$y - y_n = m(x - x_n)$$

(eq of Tangent passing through $(x_n, f(x_n))$).

new points is $(x_{n+1}, \underbrace{f(x_{n+1})}_{y_{n+1}})$

slope is the derivative at $(x_n, \underbrace{f(x_n)}_{y_n})$

so $\quad m = f'(x_n)$.

$\Rightarrow$ Eq of Tangent becomes:

$$y_{n+1} - y_n = f'(x_n)\,(x_{n+1} - x_n)$$

We want $y_{n+1} = 0$ (to get the root).

$$\cancel{-y_n = f'(x_n}$$

$$-f(x_n) = f'(x_n)\,(x_{n+1} - x_n)$$

$$-\frac{f(x_n)}{f'(x_n)} = x_{n+1} - x_n$$

$$x_n - \frac{f(x_n)}{f'(x_n)} = x_{n+1}$$

$$\boxed{x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}}$$

Ex: You want to find $\sqrt{2}$.

Let $x = \sqrt{2}$ or $x^2 = 2$

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

$f'(x) = 2x$

$$x_{n+1} = x_n - \frac{x_n^2 - 2}{2x_n}$$

$\boxed{\sqrt{2} = 1.4142135}$

Let $x_0 = 1$

$x_1 = \frac{3}{2} = 1.5$

$x_2 = \frac{17}{12} = 1.4167\ldots$

$x_3 = \frac{577}{408} = 1.4142157\ldots$

We need derivative for this method.

Method 3: Secant Method.

As $$f'(x_n) = \frac{f(x_n) - f(x_{n-1})}{x_n - x_{n-1}}$$

so

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} = x_n - \frac{f(x_n)}{\frac{f(x_n) - f(x_{n-1})}{x_n - x_{n-1}}}$$

$$\boxed{x_{n+1} = x_n - \frac{(x_n - x_{n-1})\, f(x_n)}{f(x_n) - f(x_{n-1})}}$$

LHS involves x_{n+1}

RHS " x_n, x_{n-1}

$\Rightarrow$ We need two previous points.

Above three methods work for single variable (uni-variate or single dimension).

Newton's method in Several Dimensions.

You will have system of nonlinear equations

$$f_i(x_1, x_2, \cdots, x_n) = 0 \qquad i = 1, \cdots, n.$$

In vector form

$$\underline{f}(\underline{x}) = 0$$

[Note $f'(x)$ in 1D is replaced with $\nabla \underline{f}(\underline{x})$

$\frac{1}{f'(x)}$ " " " " $[\nabla \underline{f}(\underline{x})]^{-1}$

$$\underline{x}_{n+1} = \underline{x}_n - [\nabla \underline{f}(\underline{x}_n)]^{-1} \underline{f}(\underline{x}_n)$$

(Compare with $x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$

Ex:

$$x_1^2 + x_2^2 = 1$$
$$x_2 = \sin(x_1)$$

$$f_1(x_1, x_2) = x_1^2 + x_2^2 - 1, \quad f_2(x_1, x_2) = x_2 - \sin(x_1)$$

Jacobian Matrix $\quad J = \nabla \underline{f}(\underline{x}) = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} \end{bmatrix}$

$$\frac{\partial f_1}{\partial x_1} = 2x_1, \qquad \frac{\partial f_2}{\partial x_1} = -\cos(x_1)$$

$$\frac{\partial f_1}{\partial x_2} = 2x_2, \qquad \frac{\partial f_2}{\partial x_2} = 1$$

So $J = \begin{bmatrix} 2x_1 & 2x_2 \\ -\cos(x_1) & 1 \end{bmatrix}$

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}^{-1} = \frac{1}{ad-bc} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$$

$$J^{-1} = \frac{1}{2x_1 + 2x_2 \cos(x_1)} \begin{bmatrix} 1 & -2x_2 \\ \cos(x_1) & 2x_1 \end{bmatrix}$$

The newton iteration is:

$$\begin{pmatrix} x_{1,n+1} \\ x_{2,n+1} \end{pmatrix} = \begin{pmatrix} x_{1,n} \\ x_{2,n} \end{pmatrix} - \frac{1}{2x_1 + 2x_2 \cos(x_1)} \begin{bmatrix} 1 & -2x_2 \\ \cos x_1 & 2x_1 \end{bmatrix} \begin{bmatrix} x_{1,n}^2 + x_{2,n}^2 - 1 \\ x_{2,n} - \sin(x_{1,n}) \end{bmatrix}$$

Qpla

Optimization Problems

- You want to find minimum or maximum of $F(x)$.
- We know min or max of $F(x)$ are roots of $F'(x)$ $(F'(x)=0)$
- So if we replace $f(x)$ by $F'(x)$ in the above three methods, we can find min or max of $F(x)$

So Newton's method becomes.

$$x_{n+1} = x_n - \frac{F'(x_n)}{F''(x_n)} \quad \text{(for single variable)}$$

Compare with $x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$

For multiple variables (replace $\underline{f}(\underline{x})$ with $\nabla F(\underline{x})$)

$$\underline{x}_{n+1} = \underline{x}_n - \left[\nabla\nabla \underline{f}(\underline{x}_n)\right]^{-1} \nabla F(\underline{x}_n)$$

$$= \underline{x}_n - \left[\nabla^2 F(\underline{x}_n)\right]^{-1} \nabla F(\underline{x}_n)$$

$$(\nabla\nabla F)_{i,j} = \frac{\partial^2 F}{\partial x_i \partial x_j}$$

Note: $\underline{x}_{n+1} = \underline{x}_n - \alpha \nabla F(\underline{x}_n)$ is called gradient descent (GD)

- α is a small scalar
- GD is slower than newtons method
- GD diverges from

Ex: Consider

$$F(x_1, x_2) = x_1^2 + (\log x_2)^2 \qquad x_1 \in \mathbb{R},\ x_2 > 0$$

$$\nabla F = \begin{pmatrix} \frac{\partial F}{\partial x_1} \\ \frac{\partial F}{\partial x_2} \end{pmatrix} = \begin{pmatrix} 2x_1 \\ \frac{2 \log x_2}{x_2} \end{pmatrix}$$

$$\nabla \nabla F = \begin{pmatrix} \frac{\partial^2 F}{\partial x_1^2} & \frac{\partial^2 F}{\partial x_1 \partial x_2} \\ \frac{\partial^2 F}{\partial x_2 \partial x_1} & \frac{\partial^2 F}{\partial x_2^2} \end{pmatrix} = \begin{pmatrix} 2 & 0 \\ 0 & \frac{2 - 2\log x_2}{x_2^2} \end{pmatrix}$$

$$(\nabla \nabla F)^{-1} = \frac{1}{2 \cdot \frac{2-2\log x_2}{x_2^2}} \begin{bmatrix} \frac{2-2\log x_2}{x_2^2} & 0 \\ 0 & 2 \end{bmatrix}$$

$$= \begin{bmatrix} \frac{1}{2} & 0 \\ 0 & \frac{x_2^2}{2 - 2\log x_2} \end{bmatrix}$$

Newton's iteration is:

$$\begin{pmatrix} x_{1,n+1} \\ x_{2,n+1} \end{pmatrix} = \begin{pmatrix} x_{1,n} \\ x_{2,n} \end{pmatrix} - \begin{pmatrix} \frac{1}{2} & 0 \\ 0 & \frac{x_{2,n}^2}{2 - 2\log x_{2,n}} \end{pmatrix} \begin{pmatrix} 2x_{1,n} \\ 2 \cdot \frac{\log x_{2,n}}{x_{2,n}} \end{pmatrix}$$