Carnegie Mellon

18-660: Numerical Methods for Engineering Design and Optimization

Xin Li
Department of ECE
Carnegie Mellon University
Pittsburgh, PA 15213

Electrical & Computer ENGINEERING

Overview

- Nonlinear Equation Solver
  - Newton-Raphson method
  - Binary search

Nonlinear Algebraic Equation

- Many physical systems are nonlinear and must be mathematically described as a nonlinear equation

$$F(x) = 0$$

- Example: nonlinear ordinary differential equation

$$F\left[\dot{x}(t), x(t), u(t)\right] = 0 \quad x(0) = X$$

$x(t)$: N-dimensional vector of unknown variables

$u(t)$: Vector of input sources

F : Nonlinear operator

X : Initial condition

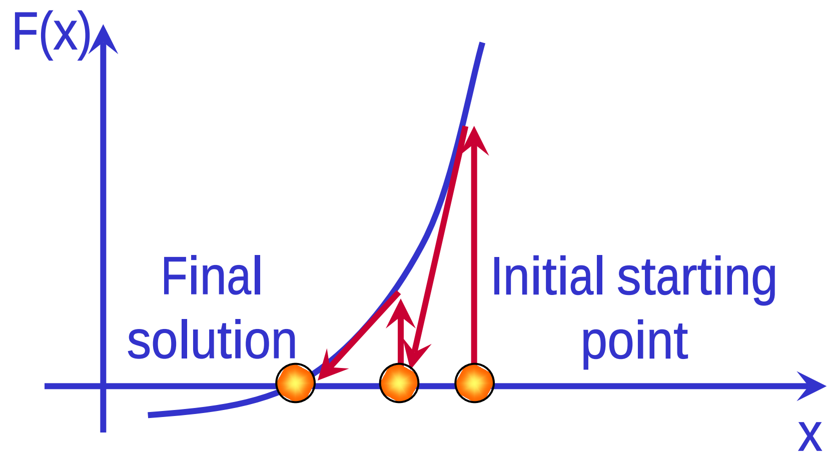
Nonlinear Algebraic Equation

$$F(x) = 0$$

- Closed-form solution cannot be derived for a general nonlinear equation
- Iterative algorithm must be applied to find an approximate solution (i.e., numerical solution)
- Newton-Raphson method is a widely-used algorithm to solve nonlinear algebraic equation

Newton-Raphson Method

- A one-dimensional example



$$F(x) = 0 \Rightarrow \begin{cases} \text{k-th iteration:} & F\left[x^{(k)}\right] \neq 0 \\ \text{Update solution:} & x^{(k+1)} = x^{(k)} + \Delta x^{(k)} \end{cases}$$

Linearize F(x) via first-order Taylor series expansion and solve iteratively

Newton-Raphson Method

- Taylor series approximation
  - For a convergent series and sufficiently small $\Delta x^{(k)}$

$$F\left[x^{(k)}+\Delta x^{(k)}\right]=F\left[x^{(k)}\right]+F'\left[x^{(k)}\right]\cdot \Delta x^{(k)}+\cdots$$

- Newton-Raphson method relies on 1st-order Taylor expansion
  - The function F(•) must be smooth and its derivative exists

Newton-Raphson Method

- Apply first-order Taylor expansion

$$F\left[x^{(k+1)}\right] \approx F\left[x^{(k)}\right] + F'\left[x^{(k)}\right] \cdot \underbrace{\left[x^{(k+1)} - x^{(k)}\right]}_{\Delta x^{(k)}} = 0$$

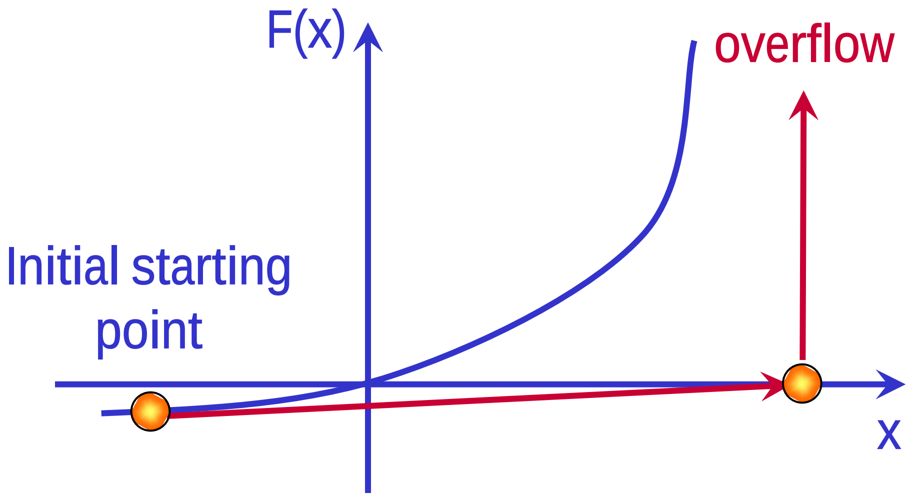
$$\Delta x^{(k)} = -\frac{F\left[x^{(k)}\right]}{F'\left[x^{(k)}\right]}$$

$$x^{(k+1)} = x^{(k)} - \frac{F\left[x^{(k)}\right]}{F'\left[x^{(k)}\right]}$$

- Global convergence is not guaranteed in general
  - Many techniques exist to improve convergence

Newton-Raphson Method

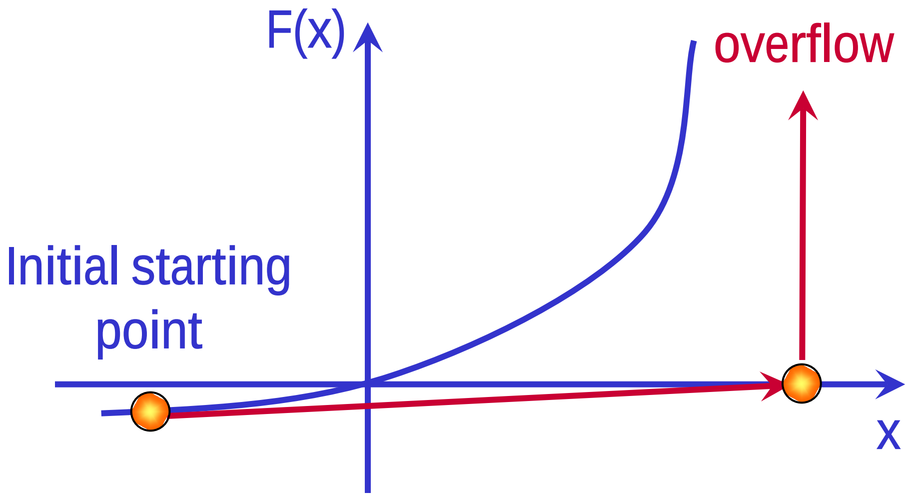
- A simple example:



$$x^{(k+1)} = x^{(k)} - \frac{F\left[x^{(k)}\right]}{F'\left[x^{(k)}\right]}$$

Newton-Raphson Method

- A simple example:



$$x^{(k+1)} = x^{(k)} - \lambda \cdot \frac{F\left[x^{(k)}\right]}{F'\left[x^{(k)}\right]} \quad (0 < \lambda < 1)$$

Damping can be applied to limit the extent of the linearized projection within one iteration

Newton-Raphson Method

$$F\left[x^{(k+1)}\right] \approx F\left[x^{(k)}\right] + F'\left[x^{(k)}\right] \cdot \left[x^{(k+1)} - x^{(k)}\right] = 0$$

Accurate if $x^{(k)}$ and $x^{(k+1)}$ are close

- Start Newton-Raphson iteration from a good initial solution
- In practice, a good initial solution may be unknown
- An alternative approach is to randomly select multiple initial solutions and apply Newton-Raphson method to each of them

A Simple Example

$$F(x) = \log(x) = 0$$

$$F'(x) = \frac{1}{x}$$

- Start from an initial solution $x^{(0)} = 0.1$

$$F\left[x^{(0)}\right] = \log(0.1) = -2.30$$

$$F'\left[x^{(0)}\right] = \frac{1}{0.1} = 10$$

A Simple Example

$$F(x) = \log(x) = 0$$

$$F'(x) = \frac{1}{x}$$

$$x^{(0)} = 0.1$$

$$F\left[x^{(0)}\right] = -2.30$$

$$F'\left[x^{(0)}\right] = 10$$

- Iteration #1

$$F\left[x^{(1)}\right] \approx F\left[x^{(0)}\right] + F'\left[x^{(0)}\right] \cdot \left[x^{(1)} - x^{(0)}\right] = 0$$

$$-2.3 + 10 \cdot \left[x^{(1)} - 0.1\right] = 0$$

$$x^{(1)} = 0.33$$

$$F\left[x^{(1)}\right] = \log(0.33) = -1.11$$

$$F'\left[x^{(1)}\right] = \frac{1}{0.33} = 3$$

A Simple Example

$$F(x) = \log(x) = 0$$

$$F'(x) = \frac{1}{x}$$

$$x^{(1)} = 0.33$$

$$F\left[x^{(1)}\right] = -1.11$$

$$F'\left[x^{(1)}\right] = 3$$

- Iteration #2

$$F\left[x^{(2)}\right] \approx F\left[x^{(1)}\right] + F'\left[x^{(1)}\right] \cdot \left[x^{(2)} - x^{(1)}\right] = 0$$

$$-1.11 + 3 \cdot \left[x^{(2)} - 0.33\right] = 0$$

$$x^{(2)} = 0.70$$

$$F\left[x^{(2)}\right] = \log(0.70) = -0.36$$

$$F'\left[x^{(2)}\right] = \frac{1}{0.70} = 1.43$$

A Simple Example

$$F(x) = \log(x) = 0$$

$$F'(x) = \frac{1}{x}$$

$$x^{(2)} = 0.70$$

$$F\left[x^{(2)}\right] = -0.36$$

$$F'\left[x^{(2)}\right] = 1.43$$

- Iteration #3

$$F\left[x^{(3)}\right] \approx F\left[x^{(2)}\right] + F'\left[x^{(2)}\right] \cdot \left[x^{(3)} - x^{(2)}\right] = 0$$

$$-0.36 + 1.43 \cdot \left[x^{(3)} - 0.70\right] = 0$$

$$x^{(3)} = 0.95$$

$$F\left[x^{(3)}\right] = \log(0.95) = -0.05$$

$$F'\left[x^{(3)}\right] = \frac{1}{0.95} = 1.05$$

Newton-Raphson Method

- Extend to multi-dimensional case

$$F(x)=0 \quad \begin{cases} F_1(x)=0 \\ F_2(x)=0 \\ \vdots \end{cases}$$

  - Apply first-order Taylor expansion for all F_i's

$$F_1\left[x^{(k+1)}\right]=F_1\left[x^{(k)}\right]+\left.\frac{\partial F_1}{\partial x_1}\right|_k \cdot\left[x_1^{(k+1)}-x_1^{(k)}\right]+\left.\frac{\partial F_1}{\partial x_2}\right|_k \cdot\left[x_2^{(k+1)}-x_2^{(k)}\right]+\cdots=0$$

$$F_2\left[x^{(k+1)}\right]=F_2\left[x^{(k)}\right]+\left.\frac{\partial F_2}{\partial x_1}\right|_k \cdot\left[x_1^{(k+1)}-x_1^{(k)}\right]+\left.\frac{\partial F_2}{\partial x_2}\right|_k \cdot\left[x_2^{(k+1)}-x_2^{(k)}\right]+\cdots=0$$

$$\vdots$$

Newton-Raphson Method

- Re-write into a matrix form

$$\begin{bmatrix} F_1\left[x^{(k+1)}\right] \\ F_2\left[x^{(k+1)}\right] \\ \vdots \end{bmatrix} = \begin{bmatrix} F_1\left[x^{(k)}\right] \\ F_2\left[x^{(k)}\right] \\ \vdots \end{bmatrix} + \begin{bmatrix} \left.\dfrac{\partial F_1}{\partial x_1}\right|_k & \left.\dfrac{\partial F_1}{\partial x_2}\right|_k & \cdots \\ \left.\dfrac{\partial F_2}{\partial x_1}\right|_k & \left.\dfrac{\partial F_2}{\partial x_2}\right|_k & \cdots \\ \vdots & \vdots & \vdots \end{bmatrix} \cdot \left(\begin{bmatrix} x_1^{(k+1)} \\ x_2^{(k+1)} \\ \vdots \end{bmatrix} - \begin{bmatrix} x_1^{(k)} \\ x_2^{(k)} \\ \vdots \end{bmatrix} \right) = 0$$

$F\left[x^{(k+1)}\right]$

$F\left[x^{(k)}\right]$

$\left.\dfrac{\partial F}{\partial x}\right|_k$ Jacobian matrix

$\Delta x^{(k)}$

Newton-Raphson Method

- The new $\Delta x^{(k)}$ can be solved by a linear solver

$$F\left[x^{(k+1)}\right] \approx F\left[x^{(k)}\right] + \left.\frac{\partial F}{\partial x}\right|_k \cdot \Delta x^{(k)} = 0$$

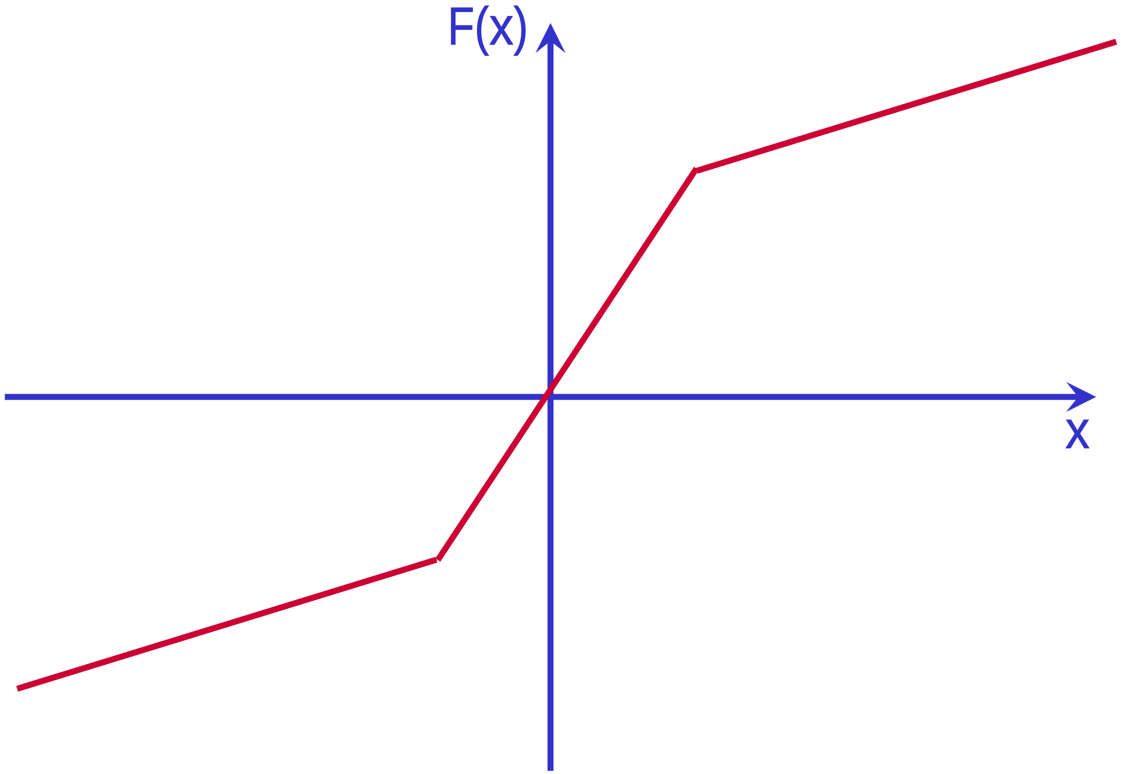
$$\Delta x^{(k)} = -\left[\left.\frac{\partial F}{\partial x}\right|_k\right]^{-1} \cdot F\left[x^{(k)}\right]$$

Vector Matrix Vector

Newton-Raphson Method

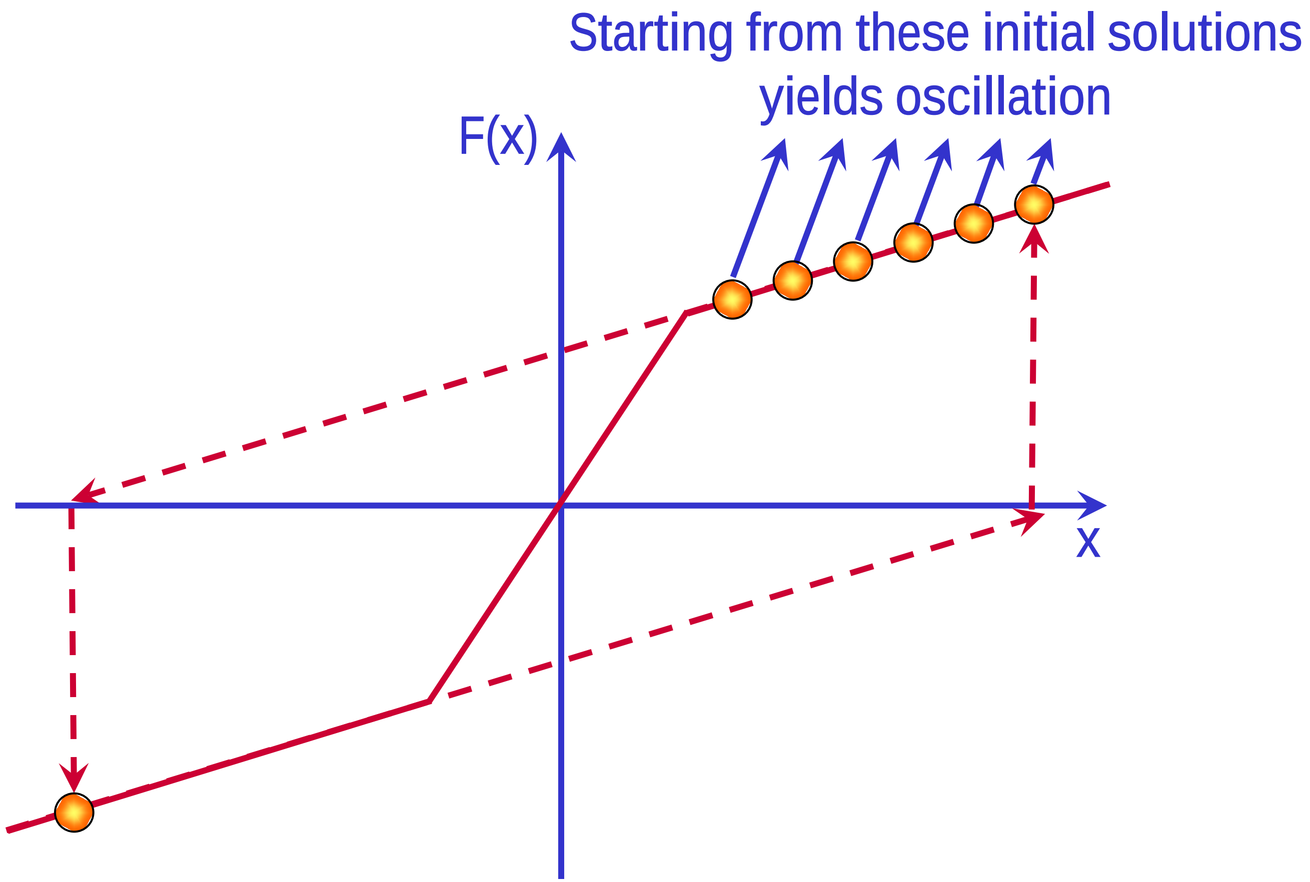
- Newton-Raphson method is applicable, if and only if F(x) is smooth and its derivative exists

- A bad example: piecewise-linear function



Newton-Raphson Method

- If we use Newton-Raphson to solve the problem
  - 2nd order derivative is not continuous



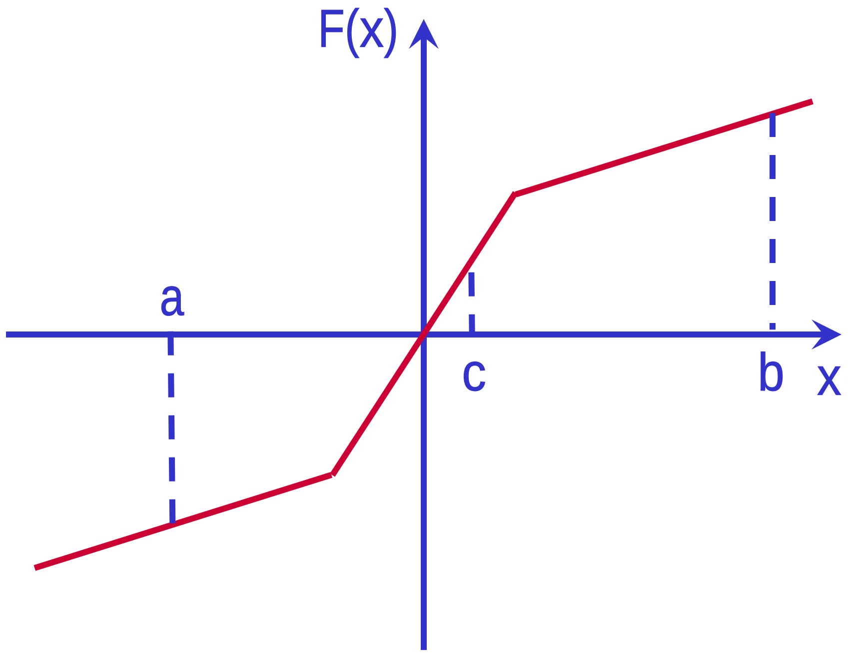
Binary Search

- Binary search is an alternative algorithm that is efficient for one-dimensional non-smooth but continuous function

- Key idea
  - Assume that the solution of F(x) = 0 is within an interval [a, b]
  - Iteratively shrink [a, b] to find the solution x

Binary Search

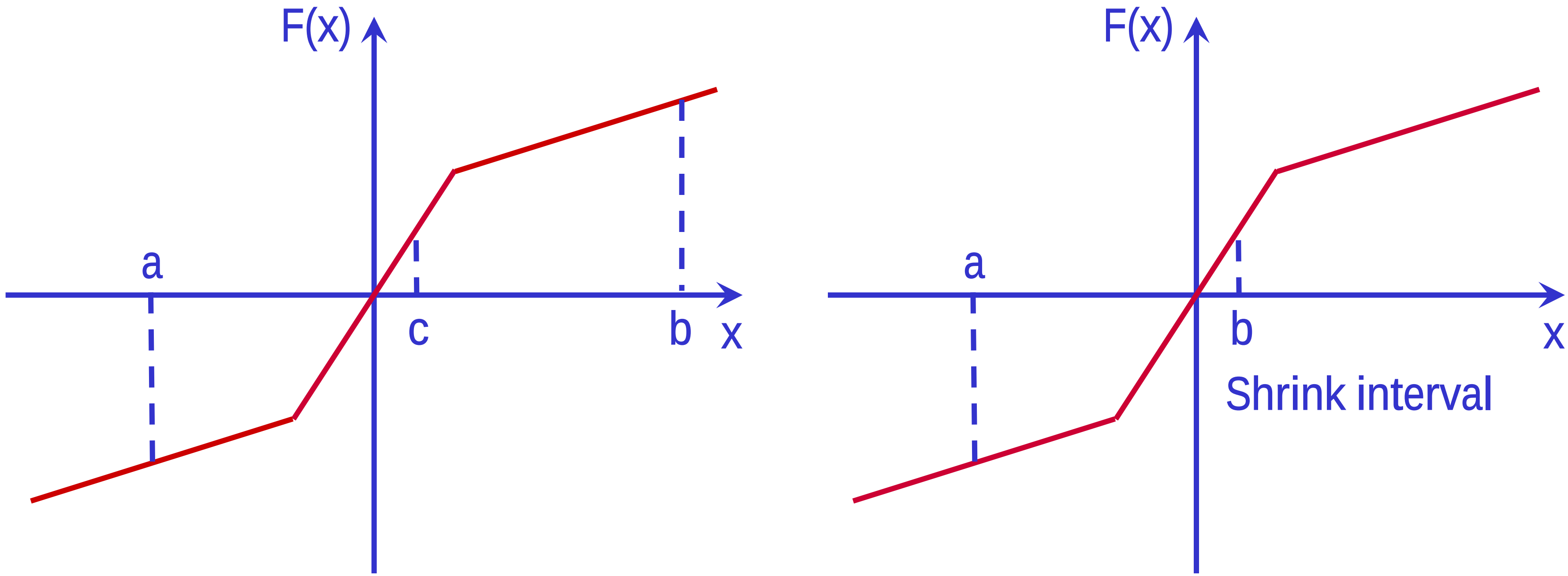
- Select initial interval [a, b] $F(a) \cdot F(b) < 0$
- Set c as the center of [a, b] $c = \frac{a+b}{2}$



Binary Search

- If $F(a)\cdot F(c)<0$
- Then $b=c$

Solution is within [a, c]

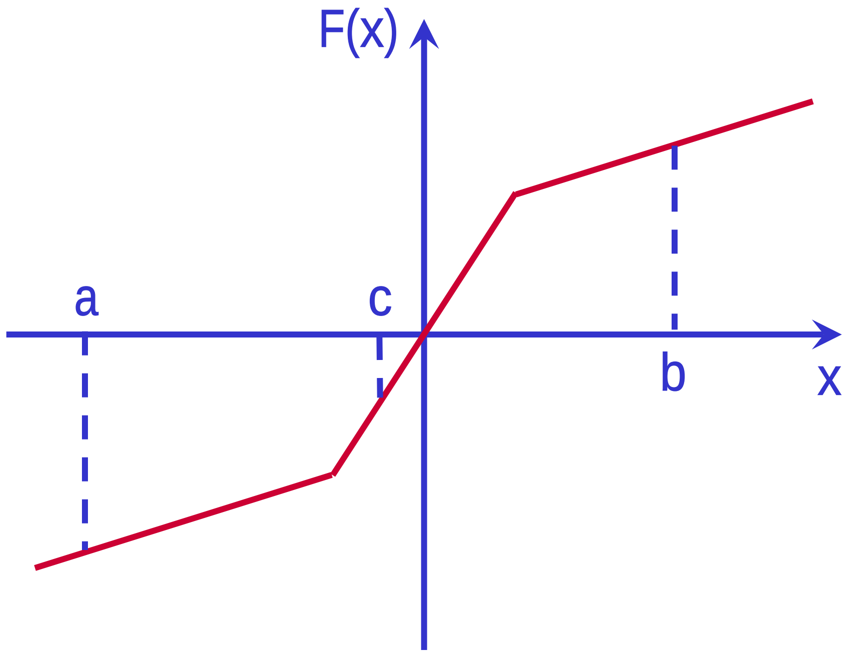


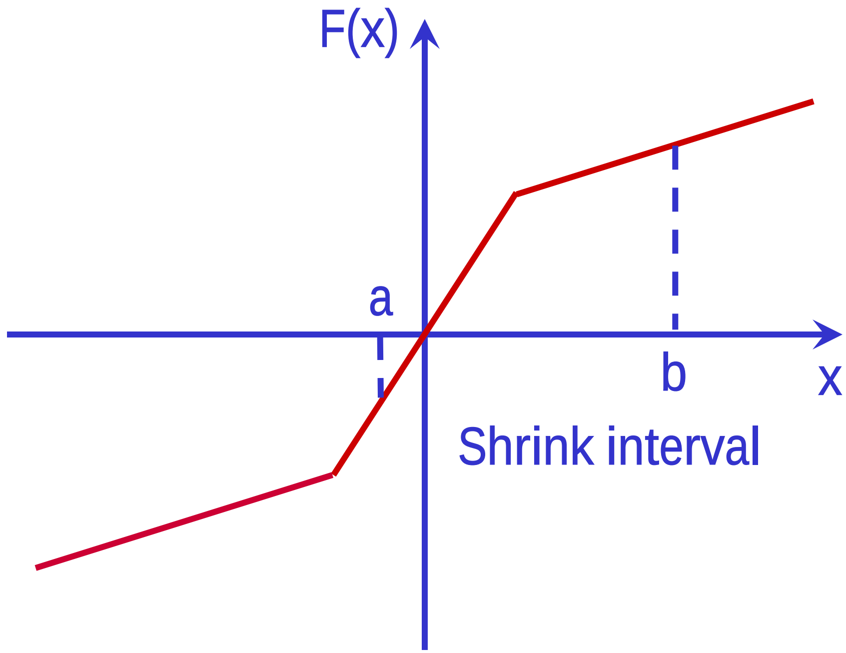
Binary Search

- If $F(b) \cdot F(c) < 0$
- Then $a = c$

Solution is within [c, b]

Continue iteration until convergence is reached





A Simple Example

$$\log(x) = 0 \quad where \quad x \in [0,10]$$

- Iteration #1

$$a = 0 \quad \log(a) < 0$$
$$b = 10 \quad \log(b) > 0$$
$$c = 5 \quad \log(c) > 0$$

$$\Rightarrow \quad x \in [0,5]$$

- Iteration #2

$$a = 0 \quad \log(a) < 0$$
$$b = 5 \quad \log(b) > 0$$
$$c = 2.5 \quad \log(c) > 0$$

$$\Rightarrow \quad x \in [0,2.5]$$

A Simple Example

$$\log(x) = 0 \quad where \quad x \in [0,10]$$

- Iteration #3

$$\begin{aligned} a &= 0 & \log(a) &< 0 \\ b &= 2.5 & \log(b) &> 0 \\ c &= 1.25 & \log(c) &> 0 \end{aligned} \quad \Rightarrow \quad x \in [0,1.25]$$

- Iteration #4

$$\begin{aligned} a &= 0 & \log(a) &< 0 \\ b &= 1.25 & \log(b) &> 0 \\ c &= 0.625 & \log(c) &< 0 \end{aligned} \quad \Rightarrow \quad x \in [0.625,1.25]$$

Continue iteration until convergence is reached

Summary

- Nonlinear equation solver
  - Newton-Raphson method
  - Binary search