Newton-Raphson method

Korea University

Dept. of Chemical & Biological Engineering

9749007

C. B. Park

NR method의 원리

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}$$

$$f'(x_i) = \frac{f(x_i)}{(x_i - x_{i+1})}$$

Tayler Series에 의한 유도

$$f(x_{i+1}) = f(x_i) + f'(x_i)(x_{i+1} - x) + \frac{f''(x_i)}{2!}(x_{i+1} - x_i)^2 + \ldots$$

0

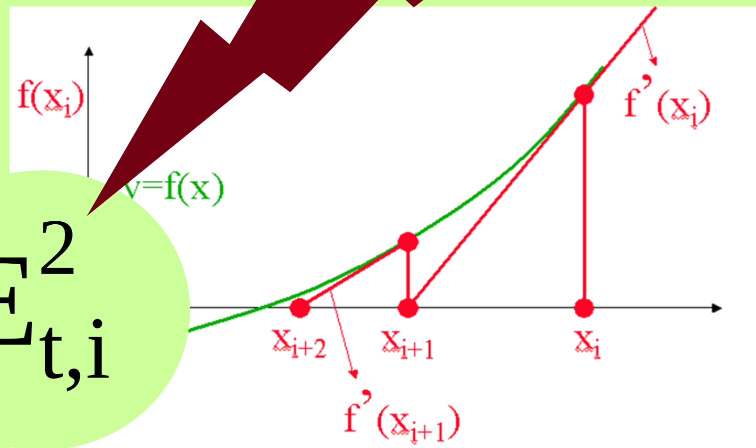
Approximation

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}$$

오차 분석

$$0 = f'(x_i)(x_r - x_{i+1}) + \frac{f''(\xi)}{2!}(x_r - x_i)^2$$

$x_i \rightarrow x_r$, $(x_r - x_{i+1}) \rightarrow E_{t,i+1}$, $\xi \rightarrow x_r$, $(x_r - x_i) \rightarrow E_{t,i}$



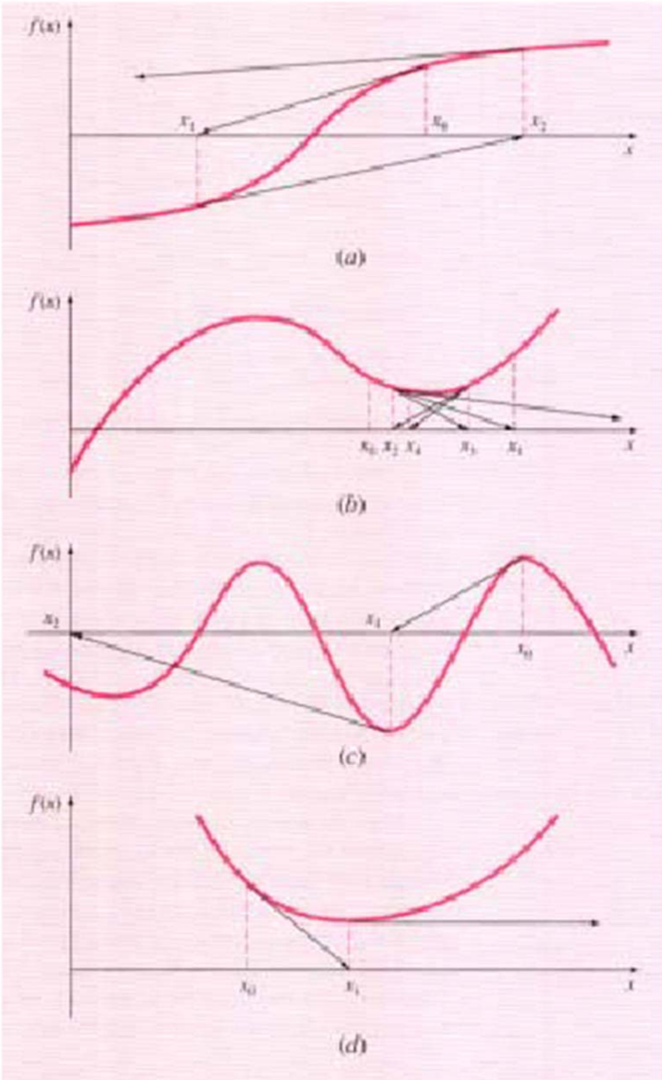
$$E_{t,i+1} = \frac{-f''(x_r)}{2f'(x_r)} E_{t,i}^2$$

```fortran
c ... f(x) = e(-x) - x
      implicit double precision (a-h,o-z)
      parameter (error=1.d-10)
      external f,fp

      print *,'initial guess of x:'
      read(*,*) x0
      iter = 1
100   delta = -f(x0)/fp(x0)
      x = x0 + delta
      print *,'iter:',iter,' delta:',delta
      if(delta.lt.error) then
      print *,'root is',x
      else
      x0 = x
      iter = iter + 1
      goto 100
      endif

      stop
      end

      double precision function f(x0)
      implicit double precision (a-h,o-z)
      f = dexp(-x0) - x0
      return
      end

      double precision function fp(x0)
      implicit double precision (a-h,o-z)
      fp = -dexp(-x0) - 1.d0
      return
      end
```

```
iter:      1 delta:  5.000000000000000E-001
iter:      2 delta:  6.631100319721815E-002
iter:      3 delta:  8.321618376440470E-004
iter:      4 delta:  1.253749188553222E-007
iter:      5 delta:  2.808428767121099E-015
root is  5.671432904097838E-001
```

(dabs(delta).lt.error)

문제점과 그 해결책



1. 가능하면 프로그램 내에 그래프 기능 포함
2. 최종 근사값 함수에 대입해서 결과가 0에 가까운지 검사
3. 프로그램의 최대 반복 횟수 제한
4. 프로그램은 계산 도중에 f’(x)=0이 되는 가능성을 검사하여 사용자에게 알려줌
5. **Modified N-R method**
   (good method for finding multiple roots)

$$u = \frac{f(x)}{f'(x)}$$

$$x^{n+1} - x^n = \delta^{n+1} = -\frac{u(x^n)}{u'(x^n)}$$

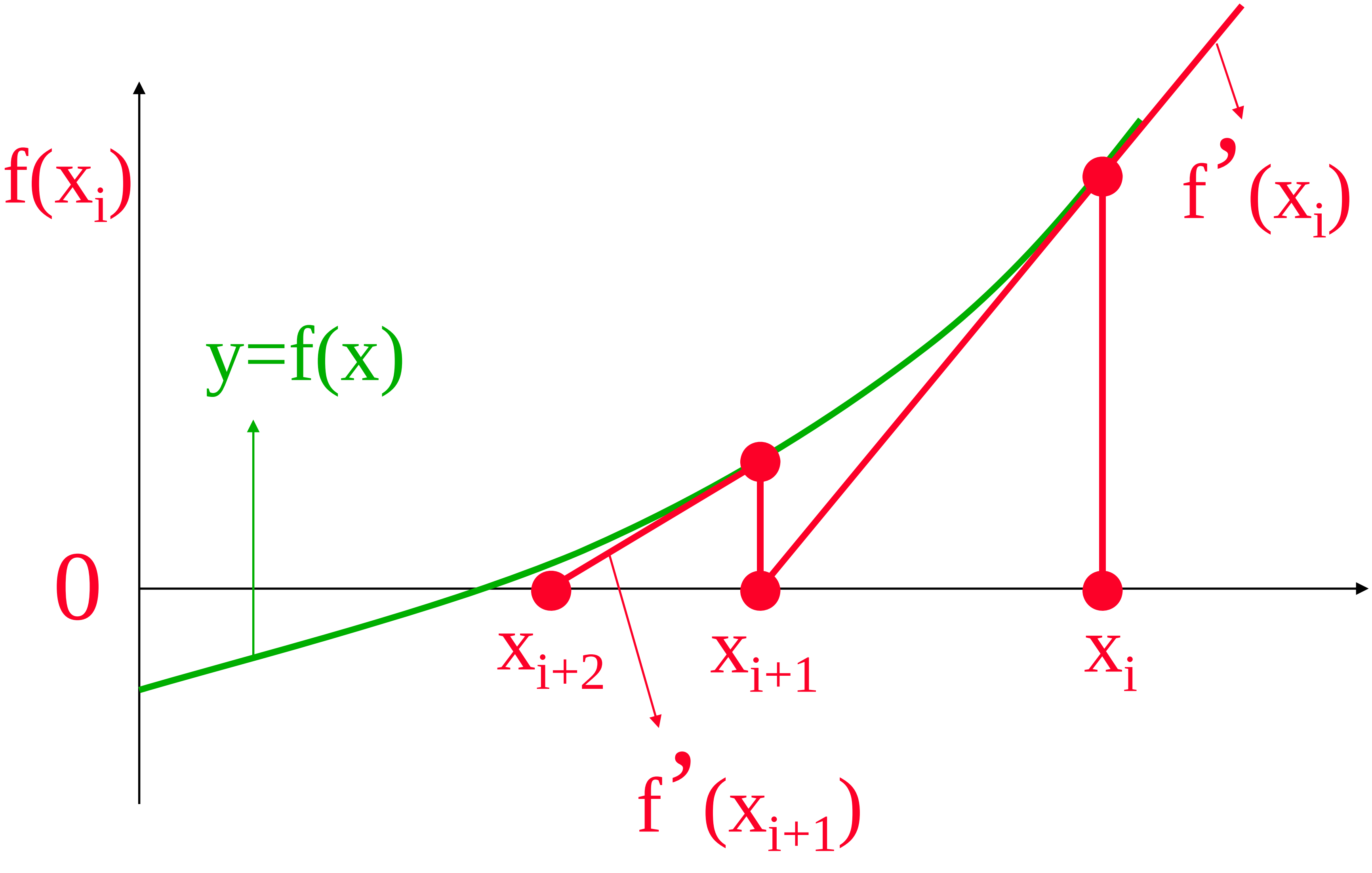
$$where \;\; u'(x) = 1 - \frac{f(x)f''(x)}{(f'(x))^2}$$

9749007

끝

박철배

NR method의 원리



NR method의 원리

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}$$

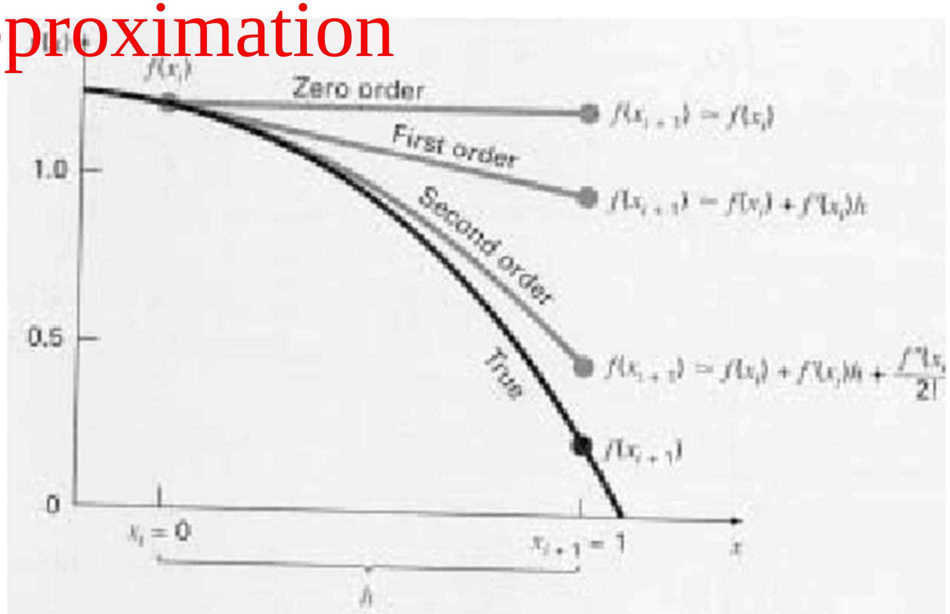
$$f'(x_i) = \frac{f(x_i)}{(x_i - x_{i+1})}$$

Tayler Series에 의한 유도

$$f(x_{i+1}) = f(x_i) + f'(x_i)(x_{i+1} - x) + \frac{f''(x_i)}{2!}(x_{i+1} - x_i)^2 + \ldots$$

0

Approximation



Tayler Series에 의한 유도

$$f(x_{i+1}) = f(x_i) + f'(x_i)(x_{i+1} - x) + \frac{f''(x_i)}{2!}(x_{i+1} - x_i)^2 + \ldots$$

0

Approximation

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}$$

오차 분석

0

$$f(x_{i+1}) = f(x_i) + f'(x_i)(x_{i+1} - x) + \frac{f''(x_i)}{2!}(x_{i+1} - x_i)^2 + \ldots$$

$$0 = f'(x_i)(x_r - x_{i+1}) + \frac{f''(\xi)}{2!}(x_r - x_i)^2$$

$$f(x_r) = f(x_i) + f'(x_i)(x_r - x) + \frac{f''(\xi)}{2!}(x_r - x_i)^2$$

0

오차 분석

$E_{t,i+1}$ $E_{t,i}$

x_r x_r

$$0 = f'(x_i)(x_r - x_{i+1}) + \frac{f''(\xi)}{2!}(x_r - x_i)^2$$

$$E_{t,i+1} = \frac{-f''(x_r)}{2f'(x_r)} E_{t,i}^2$$