

# 파트 2 방정식의 근과 최적화

---

2.1 개요

2.2 파트의 구성



## 2.1 개요 [1/2]

### ■ 근(또는 영점)이란 무엇인가?

- 2차, 3차, 4차 ... 근의 공식

예>  $f(x) = ax^2 + bx + c = 0$  에 대해  $x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$

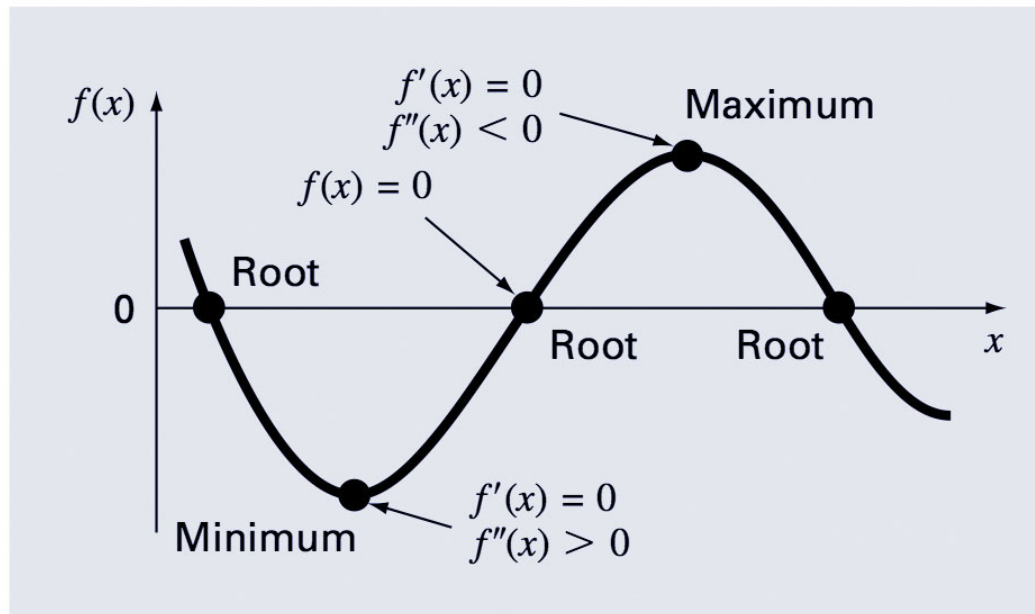
- 그래프적 방법: 함수  $f(x)$ 를 그려서  $x$ 축과 만나는 점을 찾음
  - 개략적인 추정, 정밀성의 결여
- 시행착오법:  $x$ 의 값을 가정하여  $f(x) = 0$ 이 되도록 반복하여 조정함
  - 실제 공학적 응용에 비효율적임
- 수치기법: 체계적인 방법을 통하여 근사해를 구함



## 2.1 개요 [2/2]

### ■ 최적화

- 함수의 최대값과 최소값을 구한다.



근 구하기 : 함수값이 0이 되는 위치를 찾음

최적화 : 함수의 극점을 찾음



## 2.2 파트의 구성

- 5장 : 방정식의 근 : 구간법
- 6장 : 방정식의 근 : 개방법
- 7장 : 최적화



# 5장 방정식의 근: 구간법

5.1 소개와 배경

5.2 그래프를 사용하는 방법

5.3 구간법과 초기 가정법

5.4 이분법

5.5 가위치법



# 5장 방정식의 근: 구간법

낙하속도를 구하는 문제를 다시 살펴보자.

미분방정식  $\frac{dv}{dt} = g - \frac{c_d}{m} v^2$  의 해석해

$$v(t) = \sqrt{\frac{gm}{c_d}} \tanh\left(\sqrt{\frac{gc_d}{m}} t\right)$$



(척추손상의 방지를 위해서) 항력계수가  $0.25 \text{ kg/m}$ 로 주어질 때 자유낙하 4초 후에 낙하속도가  $36 \text{ m/s}$ 를 초과하는 질량은 얼마인가?

→  $t$  대신에  $m$ 에 대해서 방정식을 표현한다!

$$f(m) = \sqrt{\frac{gm}{c_d}} \tanh\left(\sqrt{\frac{gc_d}{m}} t\right) - v(t)$$

$f(m) = 0$ 을 만족하는  $m$ 을 구하기 → "근"을 구하는 문제





# 5.1 공학과 과학에서의 근 [1/2]

## ■ 설계분야에 사용되는 기본 원리

기본원리	종 속 변 수	독립변수	매 개 변 수
열평형	온도	시간과 위치	매질의 열적 성질, 시스템의 형태
질량보존	농도 또는 질량	시간과 위치	물질의 화학적 거동, 물질전달, 시스템의 형태
힘의 평형	힘의 크기 및 방향	시간과 위치	재료의 강도, 구조적 성질, 시스템의 형태
에너지보존	운동에너지 및 포텐셜에너지	시간과 위치	매질의 열적 성질, 질량 시스템의 형태
Newton 운동법칙	가속도, 속도 및 위치	시간과 위치	질량, 시스템의 형태, 소산 매개변수
Kirchhoff 법칙	전류 및 전압	시간	전기적 성질(저항, 콘덴서, 유도자)



# 5.1 공학과 과학에서의 근 [2/2]

## ■ 외재적 표현과 내재적 표현

$$v(t) = \sqrt{\frac{gm}{c_d}} \tanh\left(\sqrt{\frac{gc_d}{m}} t\right)$$

- 매개변수와 시간이 주어지는 경우,  $v$ 를 직접 계산할 수 있다.
  - $v$ 는 외재적 값이다(식의 한쪽 변에 고립되어 있음).
- 속도, 시간, 항력계수가 주어지는 경우,  $m$ 을 직접 계산할 수 없다.

- $m$ 은 내재적 값이다.  $\Rightarrow f(m) = \sqrt{\frac{gm}{c_d}} \tanh\left(\sqrt{\frac{gc_d}{m}} t\right) - v(t) = 0$





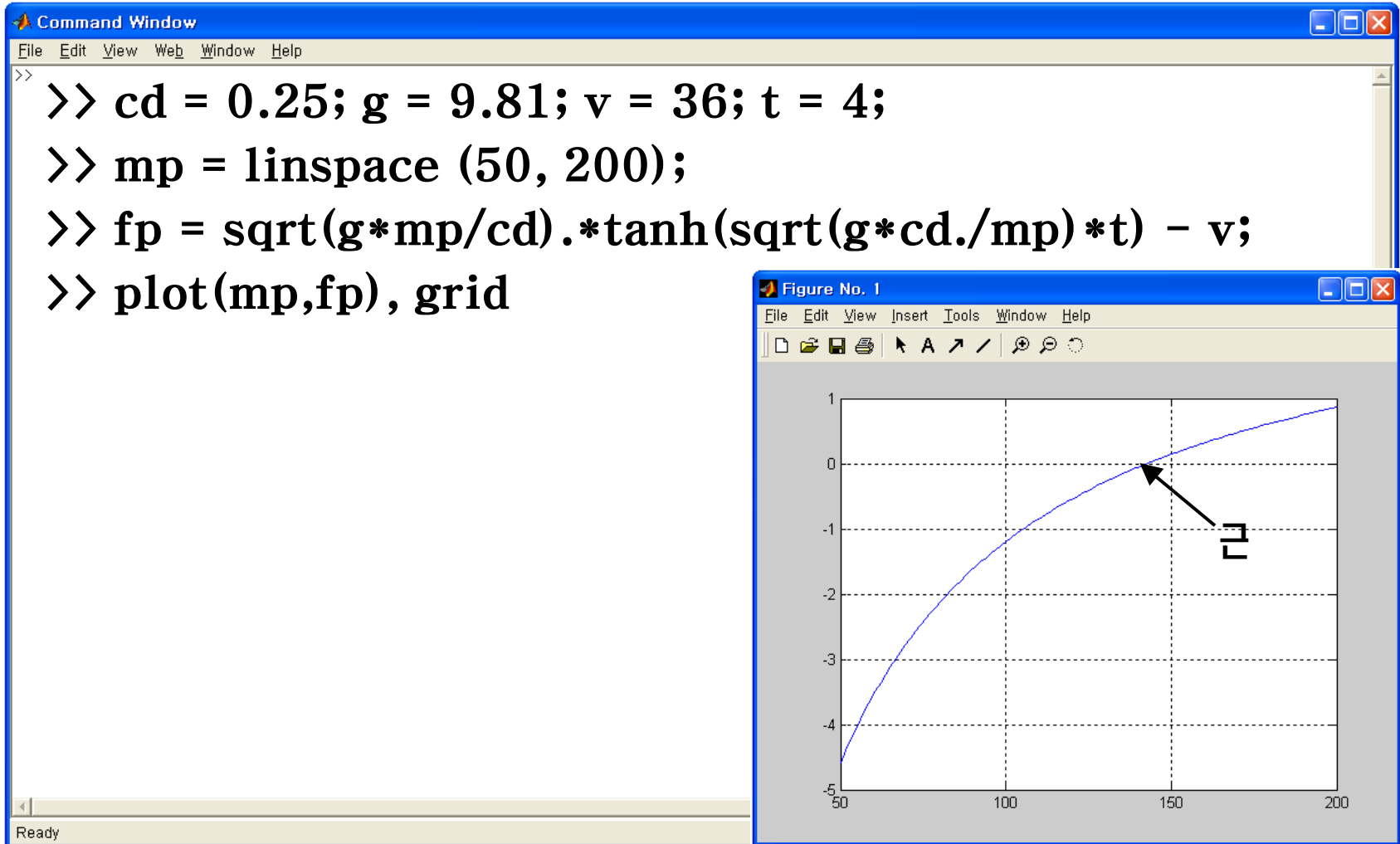
## 예제 5.1 [1/2]

Q. 자유낙하 4초 후의 속도를 36 m/s로 되게 하는 번지 점프하는 사람의 질량을 그래프적인 접근법으로 구하라.

(항력계수는 0.25 kg/m이고, 중력가속도는  $9.81 \text{ m/s}^2$ 이다.)



# 예제 5.1 [2/2]



## 예제 5.1 [2/2]

```
Command Window
File Edit View Web Window Help
>>
>> cd = 0.25; g = 9.81; v = 36; t = 4;
>> mp = linspace (50, 200);
>> fp = sqrt(g*mp/cd).*tanh(sqrt(g*cd./mp)*t) - v;
>> plot(mp,fp), grid

>> sqrt(g*145/cd)*tanh(sqrt(g*cd/145)*t) - v
ans =
    0.0456

>> sqrt(g*145/cd)*tanh(sqrt(g*cd/145)*t)
ans =
    36.0456

Ready
```



## 5.2 그래프를 사용하는 방법 (1/2)

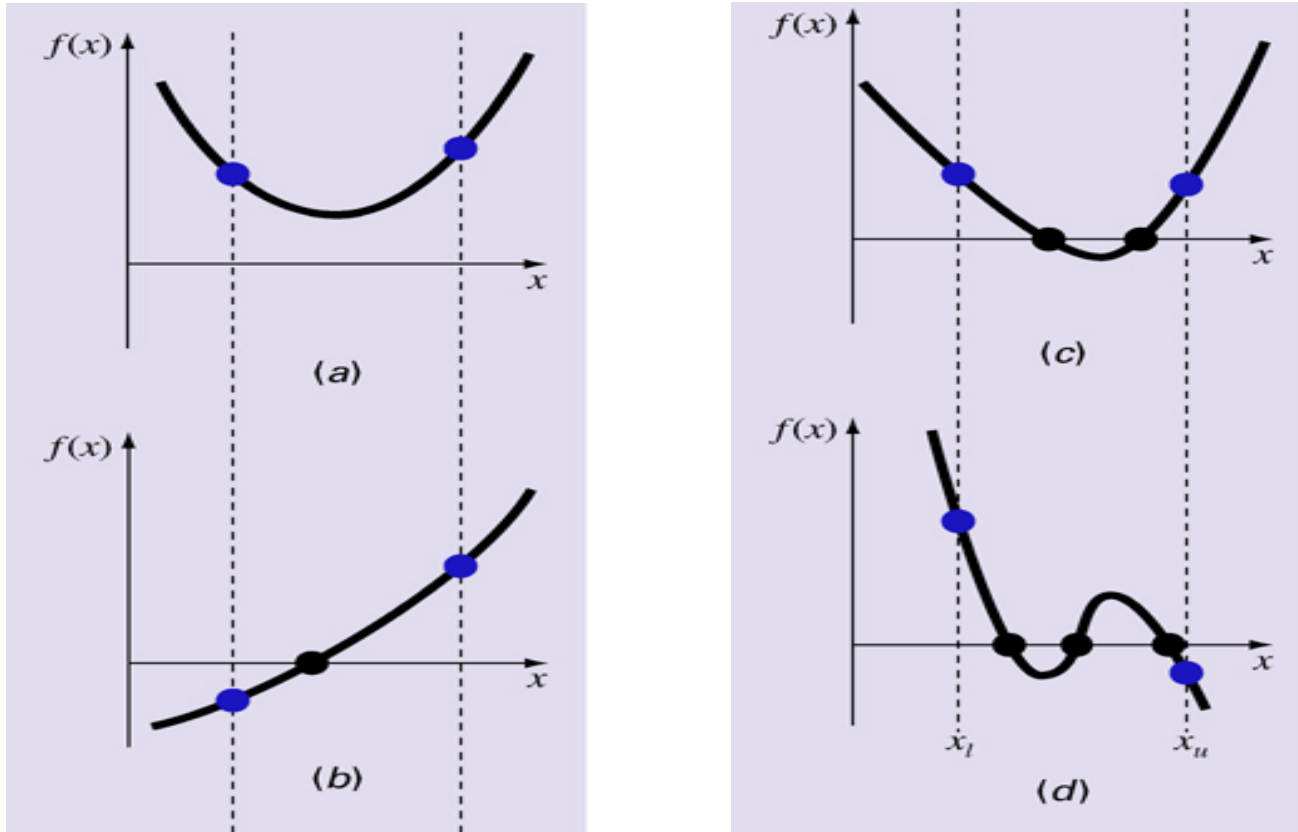


그림 5.1 하한값  $x_l$  과 상한값  $x_u$  사이의 구간에서 근이 존재할 수 있는 몇 가지 경우를 나타낸다. 구간 양끝에서 함수 값의 부호가 다르면 구간 내에 홀수 개의 근이 존재한다(b, d).



## 5.2 그래프를 사용하는 방법 (2/2)

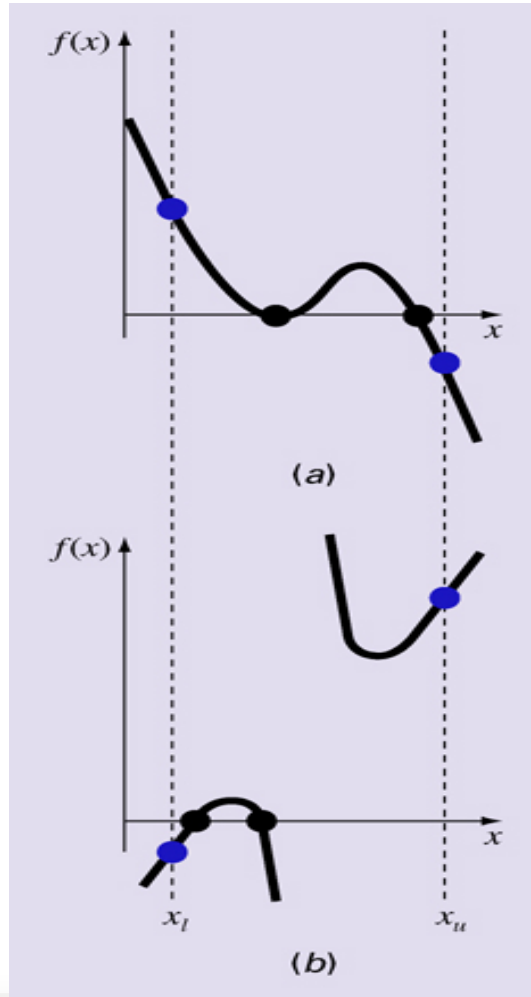


그림 5.2 중근을 갖거나 불연속 함수인 경우에는 일반적인 경우와 다르다.



## 5.3 구간법과 초기 가정법 (1/3)

### ■ 초기 가정법

#### ● 구간법

- 근을 포함하고 있는 구간의 양끝을 나타내는 초기 가정값에서부터 시작함
- 항상 근을 찾지만 수렴이 느리다

#### ● 개 방법

- 한 개 또는 그 이상의 초기 가정값에서 출발하나, 이들 값이 근을 포함할 필요는 없음
- 근을 못 찾는 경우도 있지만, 수렴이 빠르다

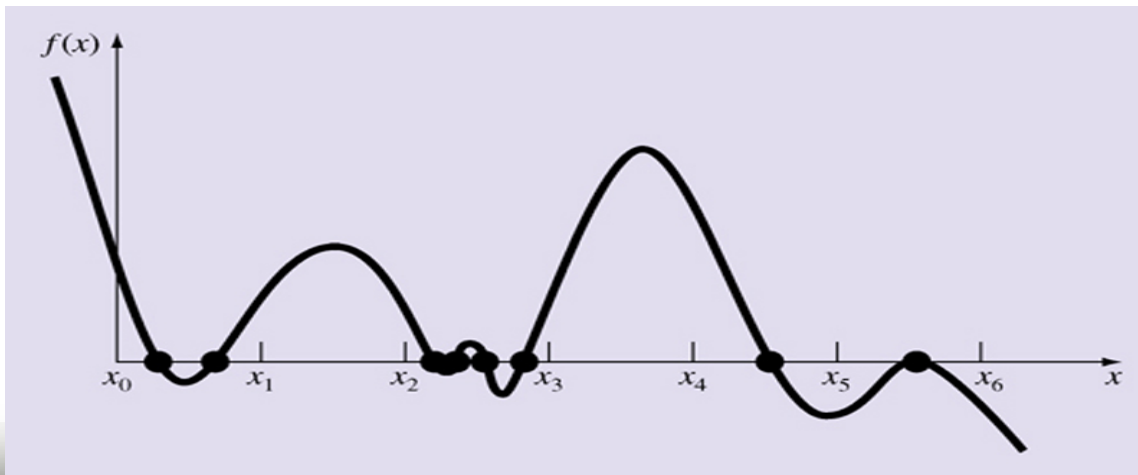




## 5.3 구간법과 초기 가정법 (2/3)

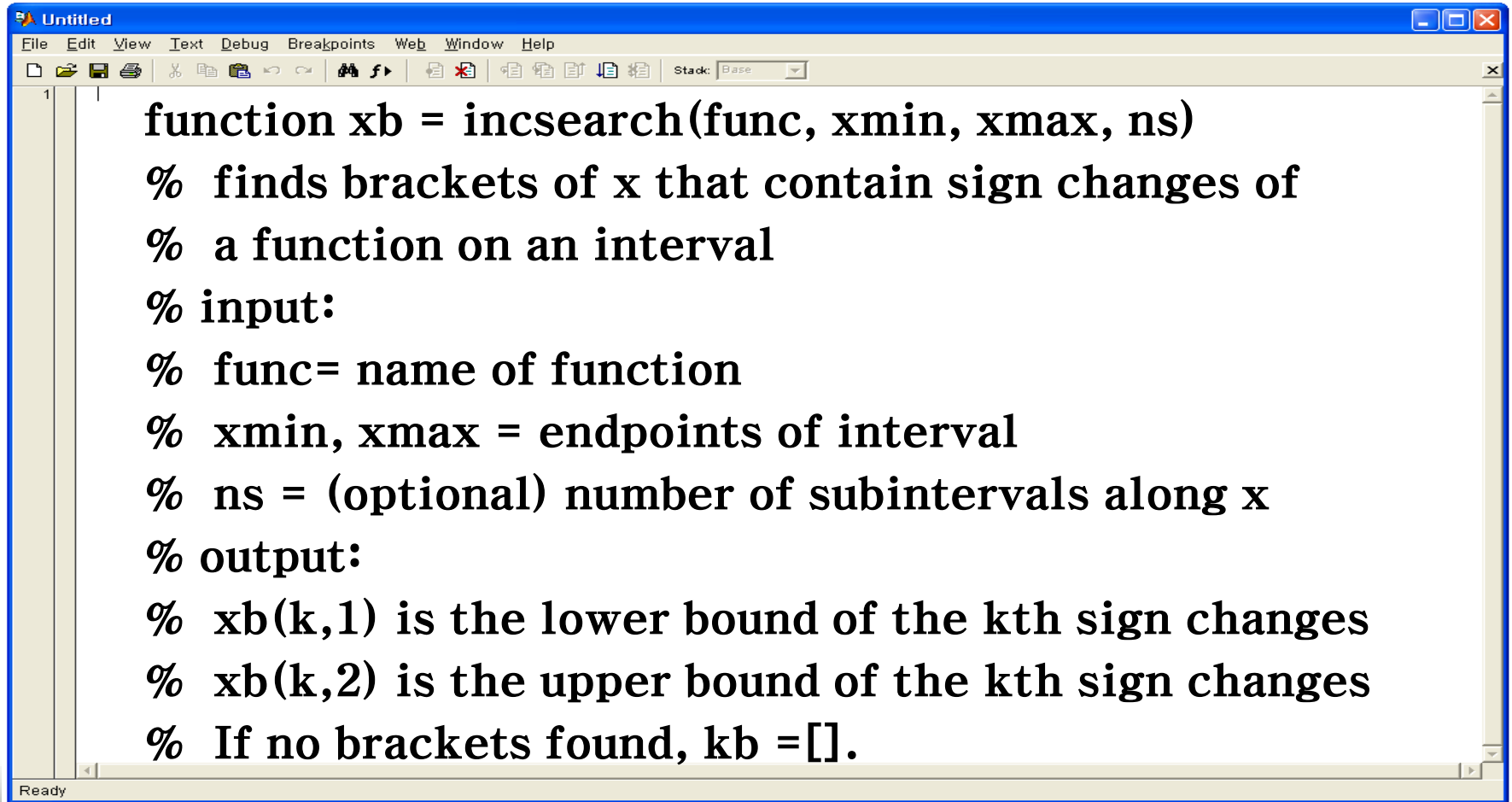
### ■ 증분탐색법

- 함수  $f(x)=0$  의 근이 존재하는 구간을 찾는다.
- $f(x_l)f(x_u) < 0$  이면 적어도  $x_l$ 과  $x_u$  사이에 실근이 하나 이상 존재한다.
- 증분 구간이 너무 작으면 계산시간이 많이 소요  
너무 크면 근을 놓치게 됨
- 증분 구간의 크기에 관계없이 중근은 놓칠 위험이 많음



## 5.3 구간법과 초기 가정법 (3/3)

### [증분탐색법을 이용하는 M-파일]

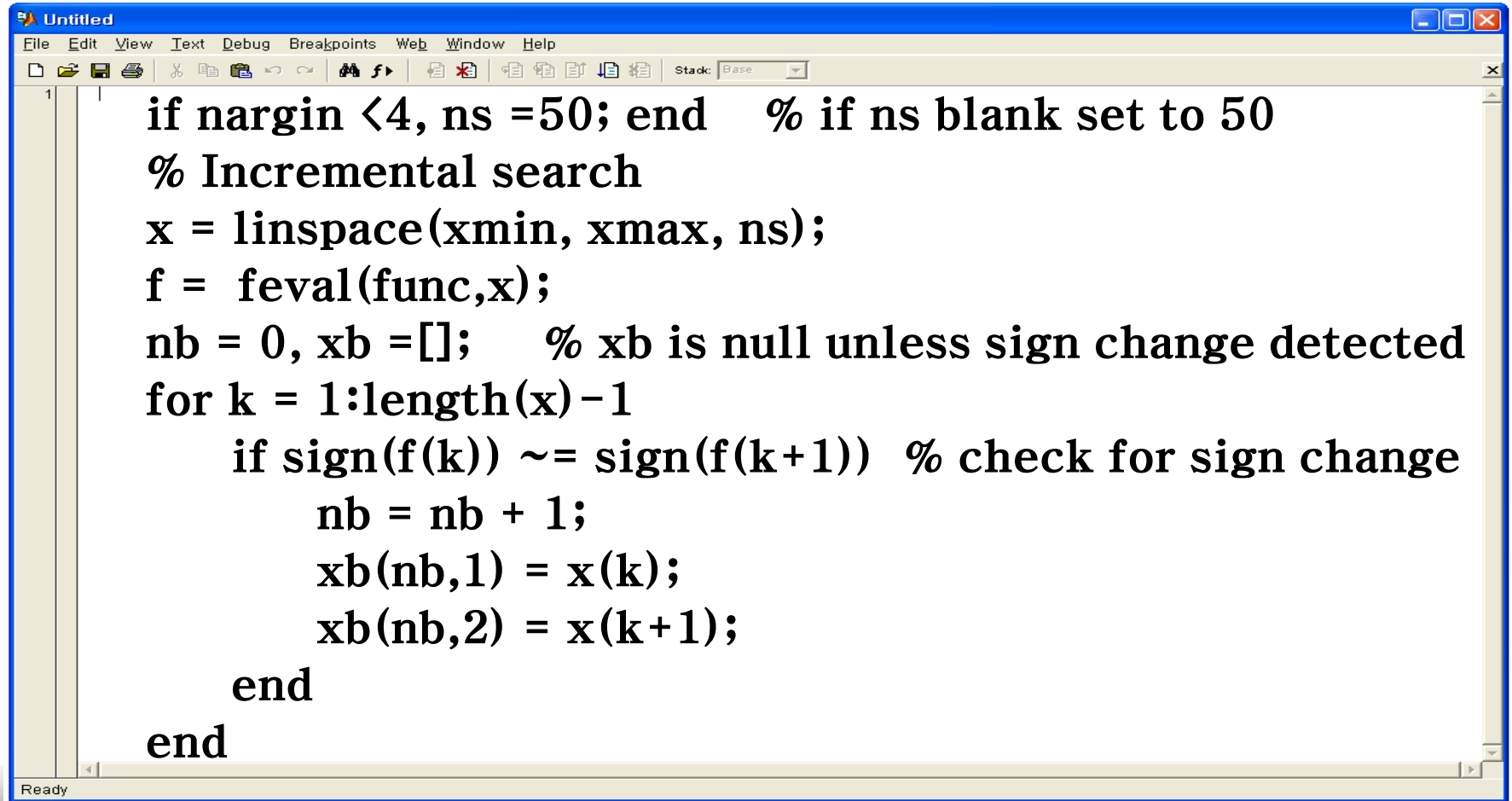


```
function xb = incsearch(func, xmin, xmax, ns)
% finds brackets of x that contain sign changes of
% a function on an interval
% input:
% func= name of function
% xmin, xmax = endpoints of interval
% ns = (optional) number of subintervals along x
% output:
% xb(k,1) is the lower bound of the kth sign changes
% xb(k,2) is the upper bound of the kth sign changes
% If no brackets found, kb = [].
```



## 5.3 구간법과 초기 가정법 (3/3)

### [증분탐색법을 이용하는 M-파일]



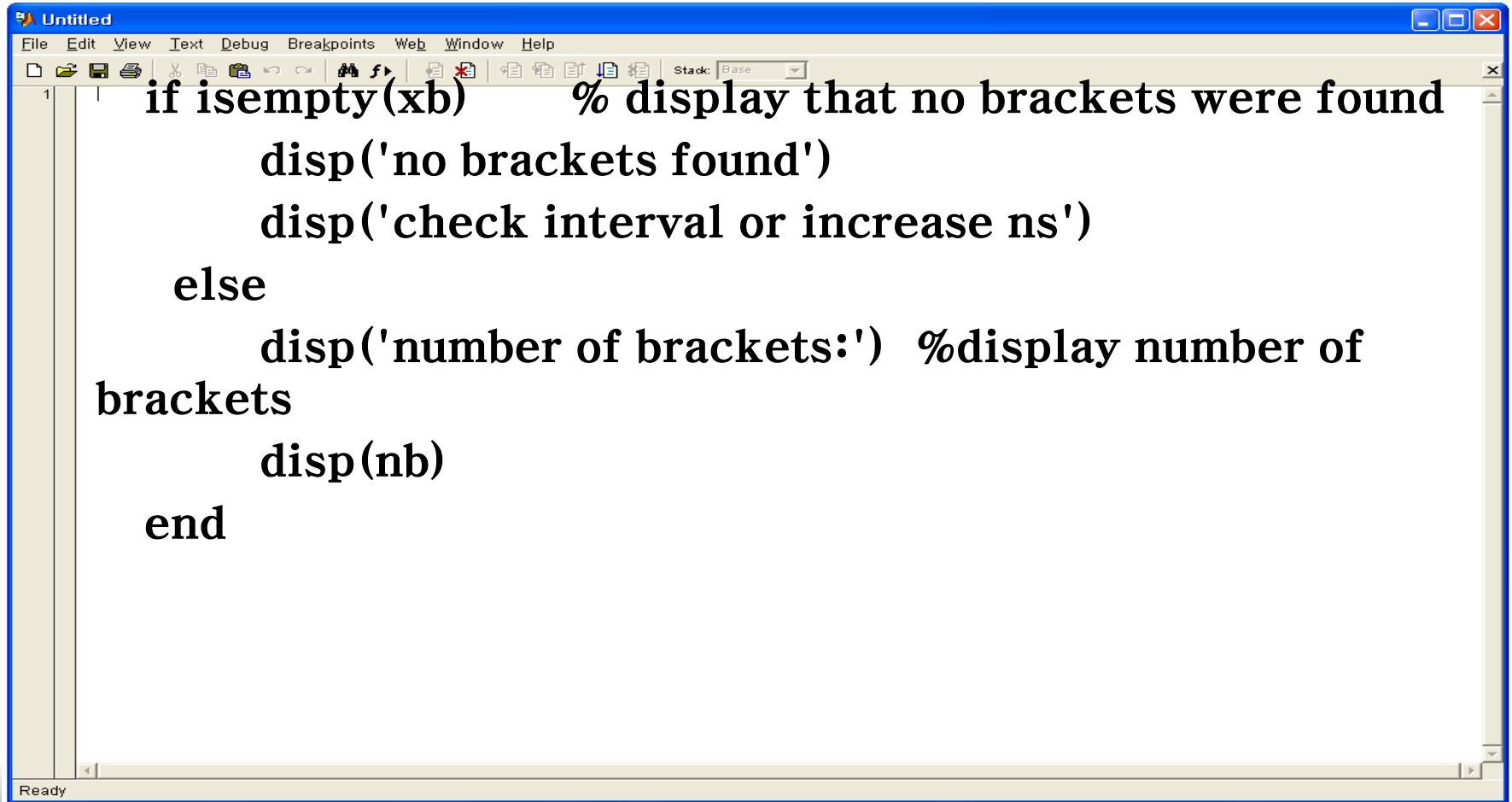
```
1  if nargin < 4, ns = 50; end    % if ns blank set to 50
    % Incremental search
    x = linspace(xmin, xmax, ns);
    f = feval(func,x);
    nb = 0, xb = [];    % xb is null unless sign change detected
    for k = 1:length(x)-1
        if sign(f(k)) ~= sign(f(k+1)) % check for sign change
            nb = nb + 1;
            xb(nb,1) = x(k);
            xb(nb,2) = x(k+1);
        end
    end
end
```

The image shows a MATLAB 'Untitled' window with a standard menu bar (File, Edit, View, Text, Debug, Breakpoints, Web, Window, Help) and a toolbar. The script is written in a light-colored editor area with a vertical line number margin on the left. The status bar at the bottom left shows 'Ready'.



## 5.3 구간법과 초기 가정법 (3/3)

[증분탐색법을 이용하는 M-파일]

A screenshot of a MATLAB 'Untitled' window. The window has a standard menu bar (File, Edit, View, Text, Debug, Breakpoints, Web, Window, Help) and a toolbar with various icons. The main text area contains a MATLAB script. The script starts with a line number '1' in the left margin. The code is as follows:

```
1 if isempty(xb) % display that no brackets were found
    disp('no brackets found')
    disp('check interval or increase ns')
else
    disp('number of brackets:') %display number of
brackets
    disp(nb)
end
```

The status bar at the bottom left of the window shows 'Ready'.

```
1 if isempty(xb) % display that no brackets were found
    disp('no brackets found')
    disp('check interval or increase ns')
else
    disp('number of brackets:') %display number of
brackets
    disp(nb)
end
```



## 예제 5.2 [1/3]

Q. `incsearch`를 사용하여 구간  $[3,6]$  사이에서  
다음 함수의 부호가 바뀌는 구간을 찾아라.

$$f(x) = \sin(10x) + \cos(3x)$$



## 예제 5.2 (2/3)

풀이)

```
Command Window
File Edit View Web Window Help
>> incsearch(inline('sin(10*x)+cos(3*x)'), 3, 6)
nb =
    0
number of brackets:
    5
ans =
    3.2449    3.3061
    3.3061    3.3673
    3.7347    3.7959
    4.6531    4.7143
    5.6327    5.6939
```

소구간이 너무 넓어서  $x=4.25$ 와  $5.2$  사이의 근을 놓쳤다. 이를 찾기 위해서 구간의 수를 다음과 같이 늘린다.





## 예제 5.2 [3/3]

```
Command Window
File Edit View Web Window Help
>> incsearch( inline('sin(10*x)+cos(3*x)'),3,6, 100)
nb =
    0
number of brackets:
    9
ans =
    3.2424    3.2727
    3.3636    3.3939
    3.7273    3.7576
    4.2121    4.2424
    4.2424    4.2727
    4.6970    4.7273
    5.1515    5.1818
    5.1818    5.2121
    5.6667    5.6970
```

Brute-force method

Ready



## 5.4 이분법 (1/7)

- 증분탐색법의 변형으로 구간 폭을 항상 반으로 나누는 방법이다.
- 함수의 부호가 구간 내에서 바뀌면 구간의 중간점에서 함수 값을 계산한다.
- 나뉜 소구간 중에서 부호가 바뀌는 소구간에 위치한 근을 구한다.
- 추정된 근의 값, 
$$x_r = \frac{x_l + x_u}{2}$$



## 5.4 이분법 (2/7)

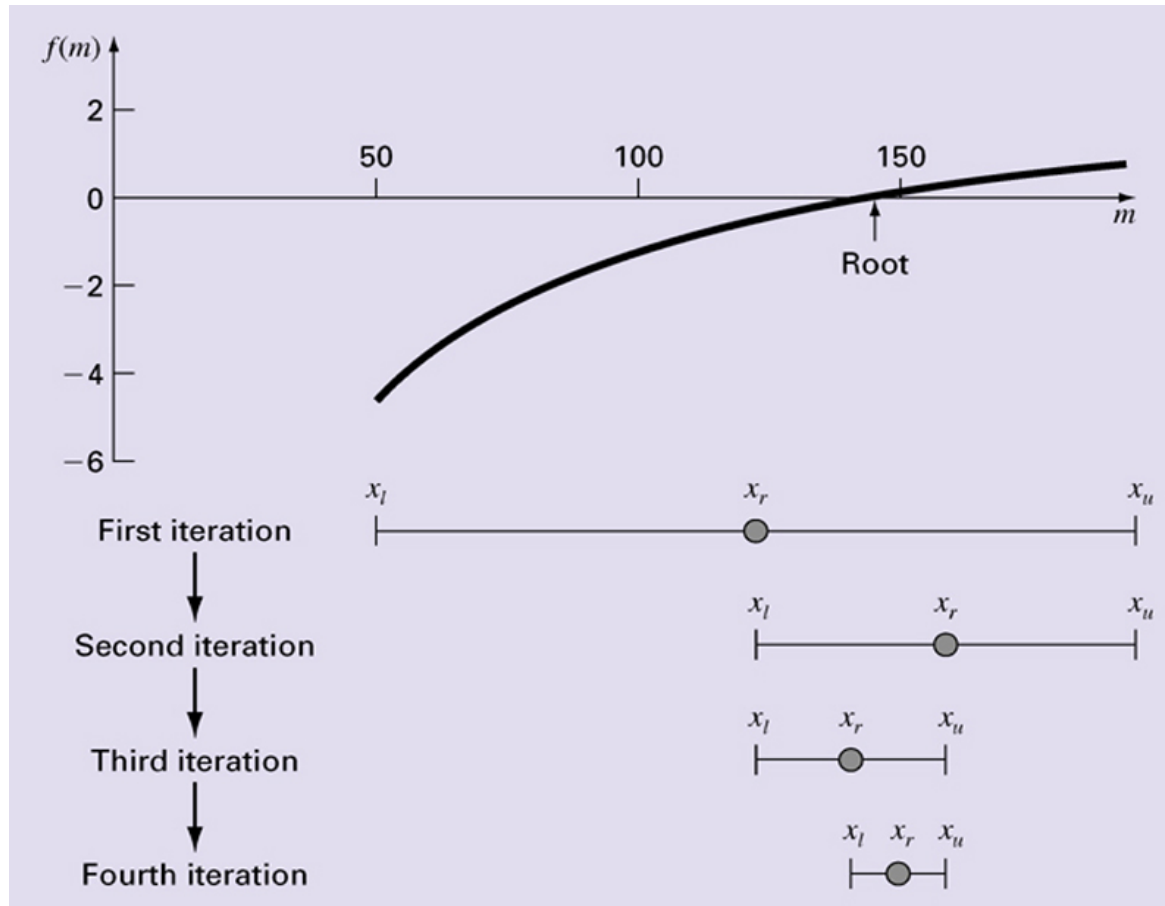


그림 5.5 이분법의 도식적 묘사.

이 그림은 예제 5.3에서 4번 반복한 것을 나타낸다.



## 5.4 이분법 (3/7)



이분법을 마치기 위한 객관적인 판단기준은?

→ 근의 참값을 모르므로  $|\varepsilon_t|$  를 이용할 수 없다.

→ 근사 상대오차,

$$|\varepsilon_a| = \left| \frac{x_r^{new} - x_r^{old}}{x_r^{new}} \right| 100\% < \varepsilon_s$$



## 예제 5.4 [1/3]

Q. 이분법을 이용하여 자유낙하 4초 후의 속도를 36 m/s로 되게 하는 번지 점프하는 사람의 질량을 구하라. 근사오차가  $\varepsilon_s = 0.5\%$ 의 종료 판정기준 이하가 될 때까지 계산을 반복하라.

단, 항력계수는 0.25 kg/m이고, 중력가속도는 9.81 m/s<sup>2</sup>이다.



# 예제 5.4 (2/3)

반 복	구간		추정 근	오차(%)		
	$x_l$	$x_u$	$x_r$	$ \epsilon_a $	$ \epsilon_t $	
1	50	200	$\frac{50+200}{2}=125$		$\left  \frac{142.7376-125}{142.7376} \right  100\% = 12.43\%$	$f(50)f(125) = -4.579(-0.409) = 1.871$
2	125	200	$\frac{125+200}{2}=162.5$	23.08	13.85	$f(125)f(162.5) = -0.409(0.359) = -0.147$
3	125	162.5	$\frac{125+162.5}{2}=143.75$	13.04	0.71	
4	125	143.75	134.375	6.98	5.86	
5	134.375	143.75	139.0625	3.37	2.58	
6	139.0625	143.75	141.4063	1.66	0.93	
7	141.4063	143.75	142.5781	0.82	0.11	
8	142.5781	143.75	143.1641	0.41	0.30	





## 예제 5.4 [3/3]

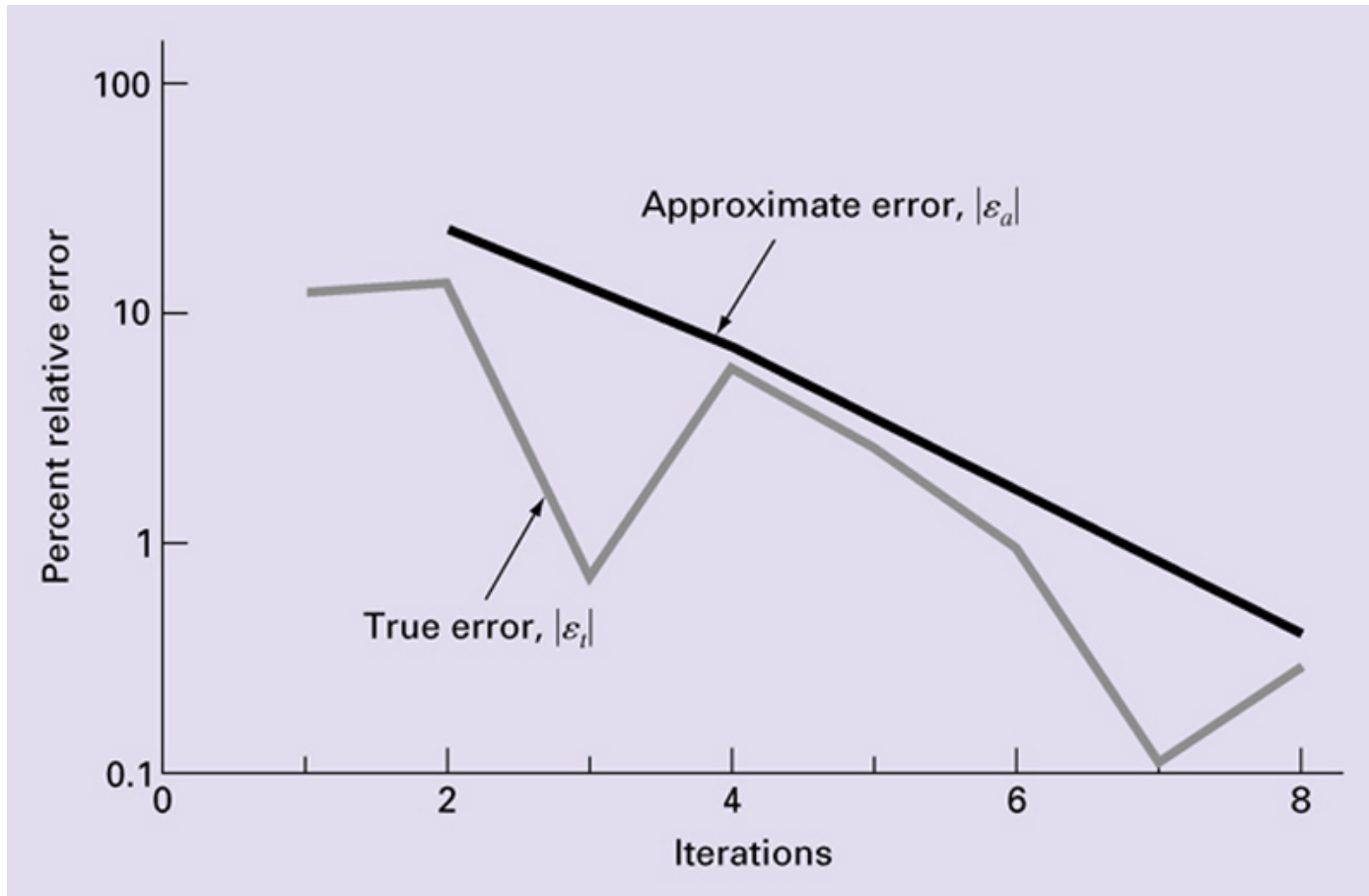


그림 5.6 이분법에서의 오차.

반복횟수에 대해 참오차와 근사오차가 그림으로 그려져 있다.



## 5.4 이분법 (4/7)



왜 참오차는 들쭉날쭉한 형태를 갖는가?

- 구간 내의 어느 점이나 참근이 될 수 있기 때문이다.
- 참근이 구간의 중앙에 위치할 때는  $|\varepsilon_t|$  와  $|\varepsilon_a|$  의 차이가 크다.
- 참근이 구간의 끝 쪽에 위치할 때는  $|\varepsilon_t|$  와  $|\varepsilon_a|$  의 차이가 작다.



## 5.4 이분법 (5/7)

- 이분법은 일반적으로 다른 방법에 비해 수렴속도가 느리다.
- $|\varepsilon_a|$ 는  $|\varepsilon_t|$ 이 줄어드는 일반적으로 추이를 나타낸다.
- $|\varepsilon_a|$ 는 참오차의 상한이므로  $|\varepsilon_t|$  보다 항상 크다.
- 절대오차,  $E_a^0 = \Delta x^0 = x_u^0 - x_l^0$       반복을 시작하기 전

$$E_a^1 = \frac{\Delta x^0}{2} \quad \text{1번 반복 후}$$

$$E_a^n = \frac{\Delta x^0}{2^n} \quad n \text{ 차례 반복 후}$$

- 만약  $E_{a,d}$  가 원하는 오차라면,  $n = \frac{\log(\Delta x^0 / E_{a,d})}{\log 2} = \log_2 \left( \frac{\Delta x^0}{E_{a,d}} \right)$

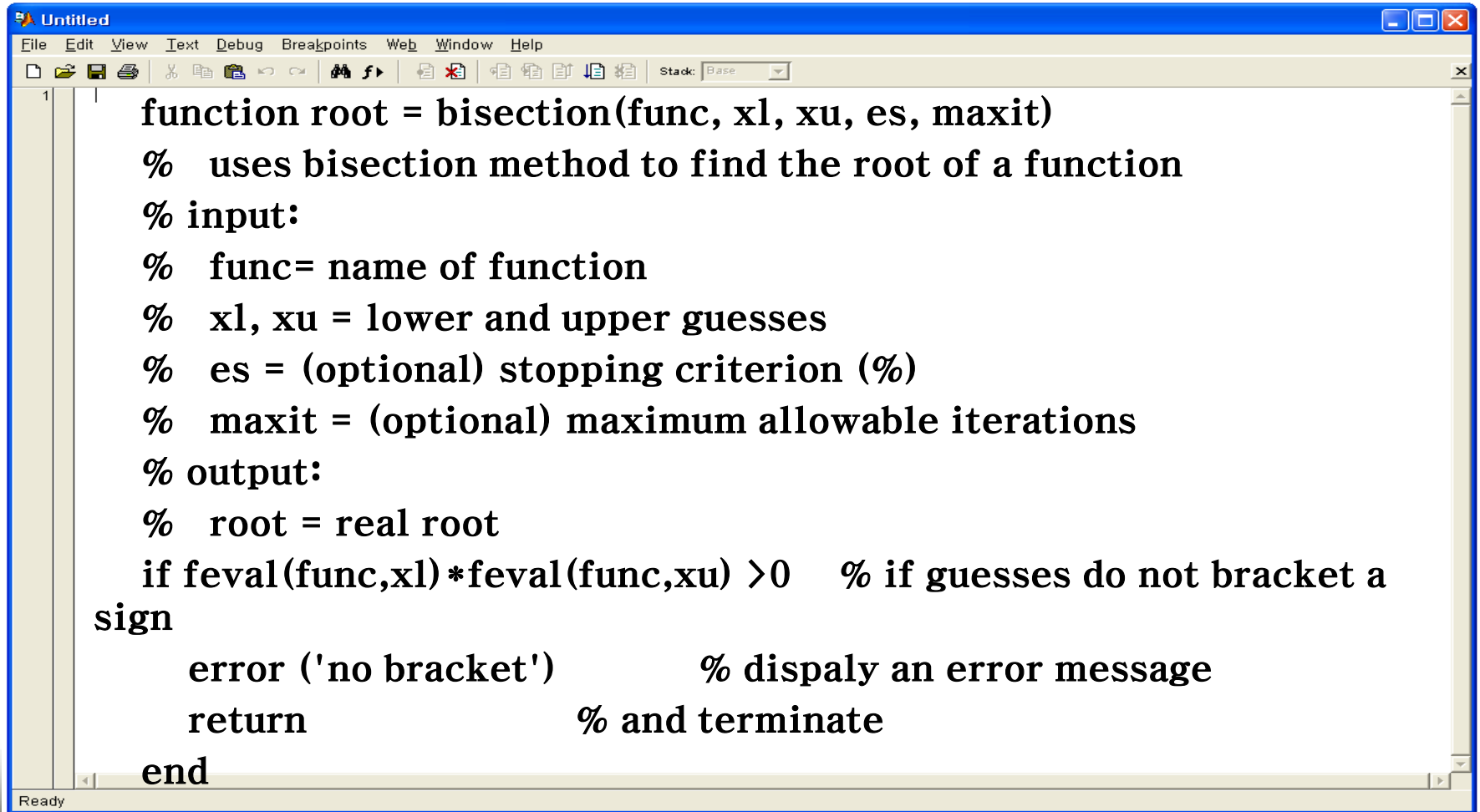
- 위의 예에서 8번 반복 후에는  $E_a = \frac{|143.7500 - 142.5781|}{2} = 0.5859$  이 된다.

- 또는  $n = \log_2 \left( \frac{150}{0.5859} \right) = 8$



## 5.4 이분법 (6/7)

### [이분법을 수행하기 위한 M-파일]

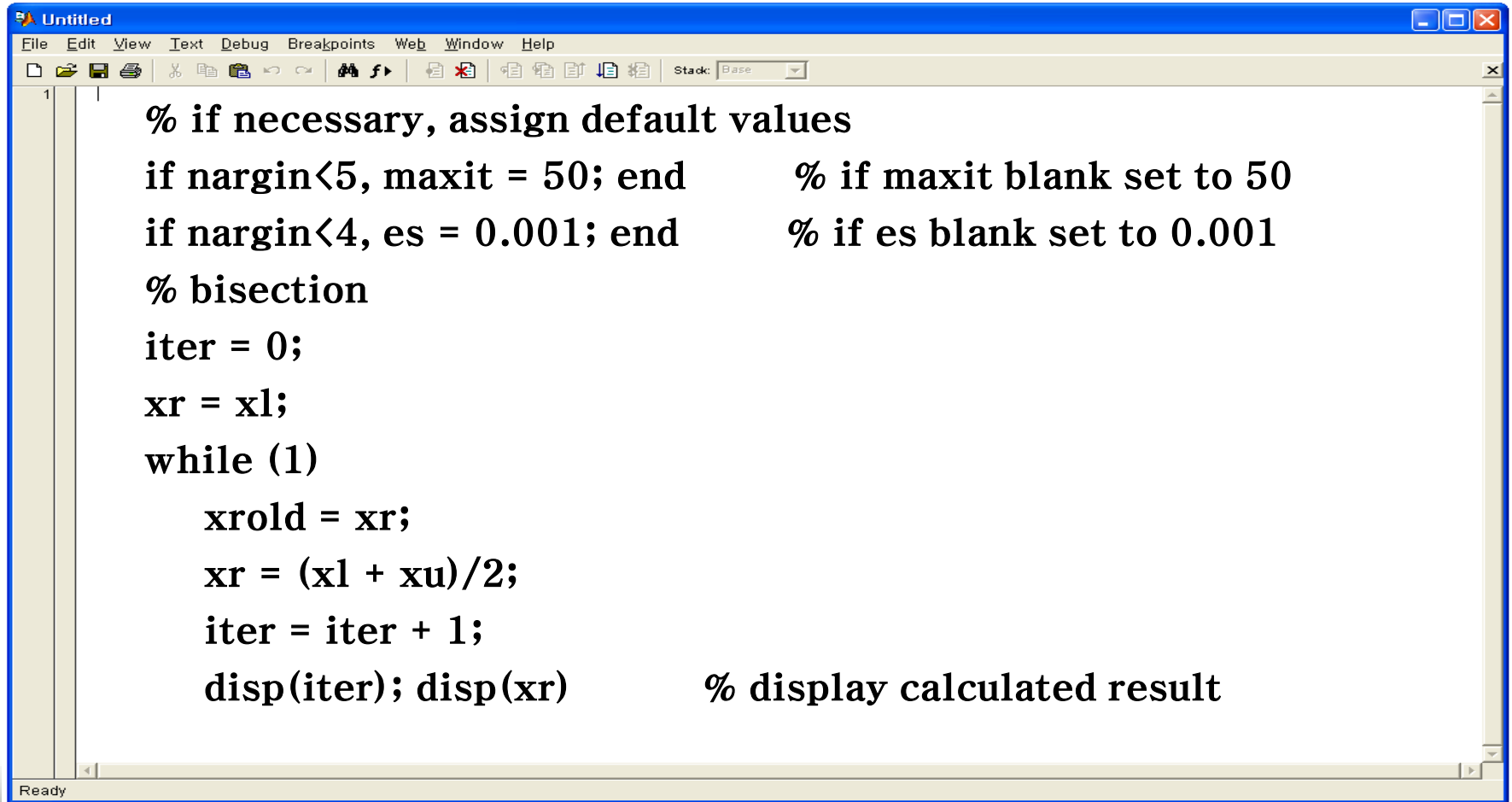


```
function root = bisection(func, xl, xu, es, maxit)
% uses bisection method to find the root of a function
% input:
% func= name of function
% xl, xu = lower and upper guesses
% es = (optional) stopping criterion (%)
% maxit = (optional) maximum allowable iterations
% output:
% root = real root
if feval(func,xl)*feval(func,xu) > 0 % if guesses do not bracket a
sign
    error('no bracket') % display an error message
    return % and terminate
end
```



## 5.4 이분법 (6/7)

### [이분법을 수행하기 위한 M-파일]

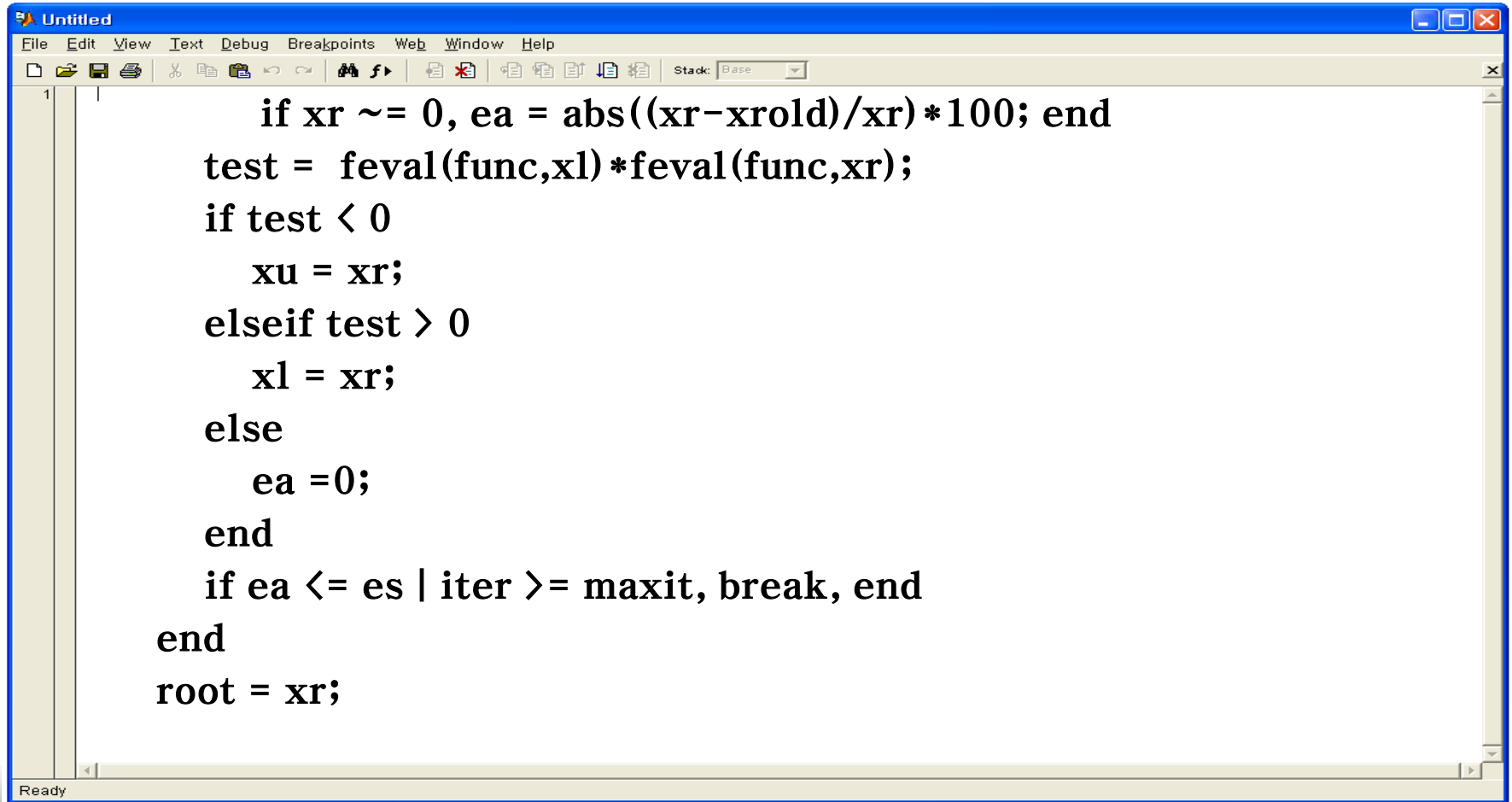
A screenshot of a MATLAB 'Untitled' window. The window has a blue title bar and a menu bar with 'File', 'Edit', 'View', 'Text', 'Debug', 'Breakpoints', 'Web', 'Window', and 'Help'. Below the menu bar is a toolbar with various icons for file operations, editing, and debugging. The main area of the window contains a script for the bisection method. The script starts with a line number '1' in the left margin. The code includes comments in Korean and English, and uses standard MATLAB syntax for conditional statements, loops, and arithmetic operations. The status bar at the bottom left shows 'Ready'.

```
1 % if necessary, assign default values
if nargin<5, maxit = 50; end      % if maxit blank set to 50
if nargin<4, es = 0.001; end    % if es blank set to 0.001
% bisection
iter = 0;
xr = xl;
while (1)
    xrold = xr;
    xr = (xl + xu)/2;
    iter = iter + 1;
    disp(iter); disp(xr)        % display calculated result
```



## 5.4 이분법 (6/7)

### [이분법을 수행하기 위한 M-파일]

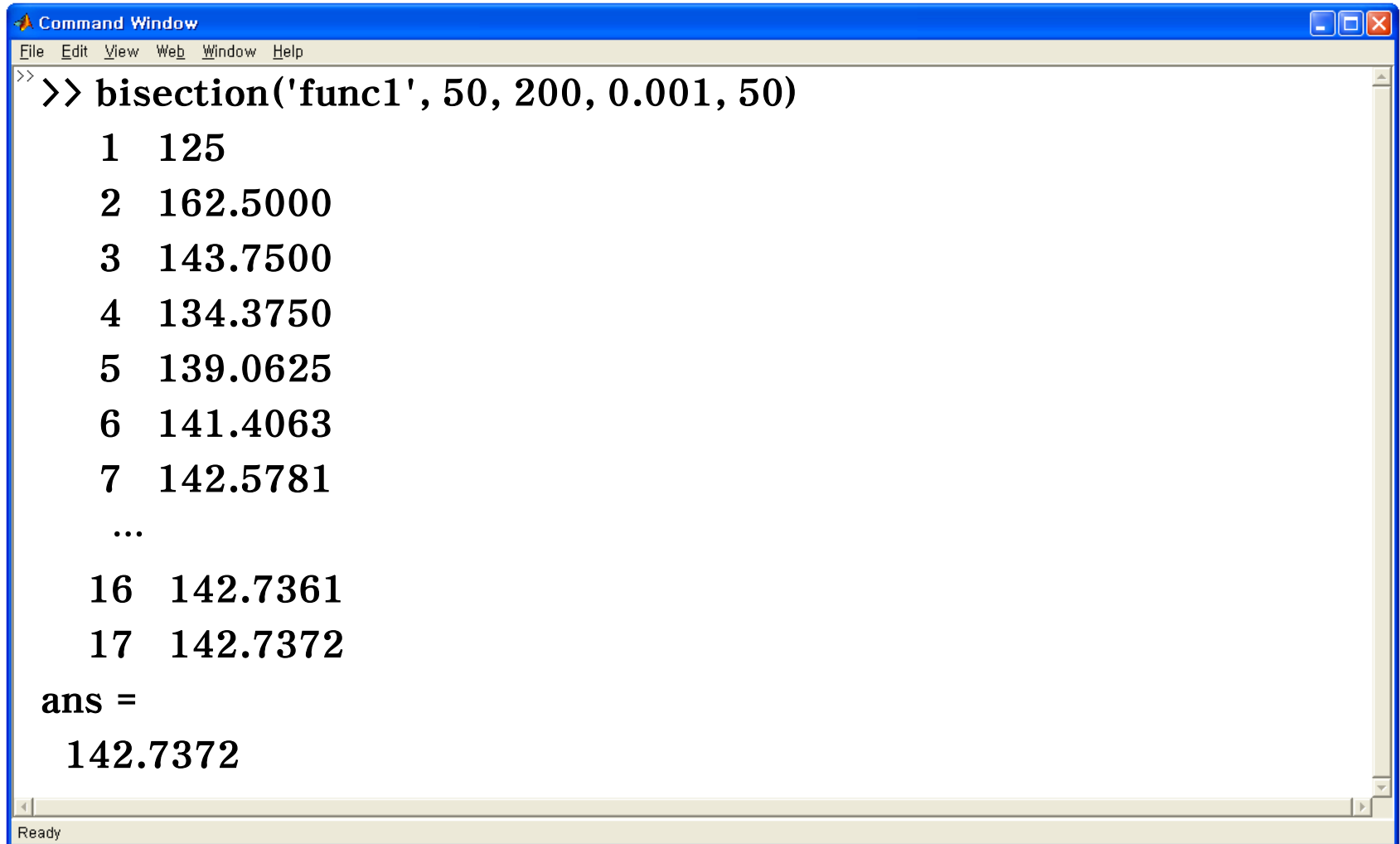


```
1      if xr ~= 0, ea = abs((xr-xrold)/xr)*100; end
test = feval(func,xl)*feval(func,xr);
if test < 0
    xu = xr;
elseif test > 0
    xl = xr;
else
    ea = 0;
end
if ea <= es | iter >= maxit, break, end
end
root = xr;
```





## 5.4 이분법 (7/7)



```
>> bisecton('func1', 50, 200, 0.001, 50)
1 125
2 162.5000
3 143.7500
4 134.3750
5 139.0625
6 141.4063
7 142.5781
...
16 142.7361
17 142.7372
ans =
142.7372
```



## 5.5 가위치법 (1/3)

- 선형보간법이라고도 하는 구간법이다.
- ◆ 이분법과 매우 유사하다.
- ◆ 구간을 반분하기보다는  $f(x_l)$ 과  $f(x_u)$ 를 연결하는 직선과  $x$ 축의 교점으로 새로운 근을 구하는 방법이다.
- ◆ 가위치법 공식

$$x_r = x_u - \frac{f(x_u)(x_l - x_u)}{f(x_l) - f(x_u)}$$



## 5.5 가위치법 [2/3]



### 가위치법 공식의 유도

닮은꼴 삼각형에서  $\frac{f(x_l)}{x_r - x_l} = \frac{f(x_u)}{x_r - x_u}$

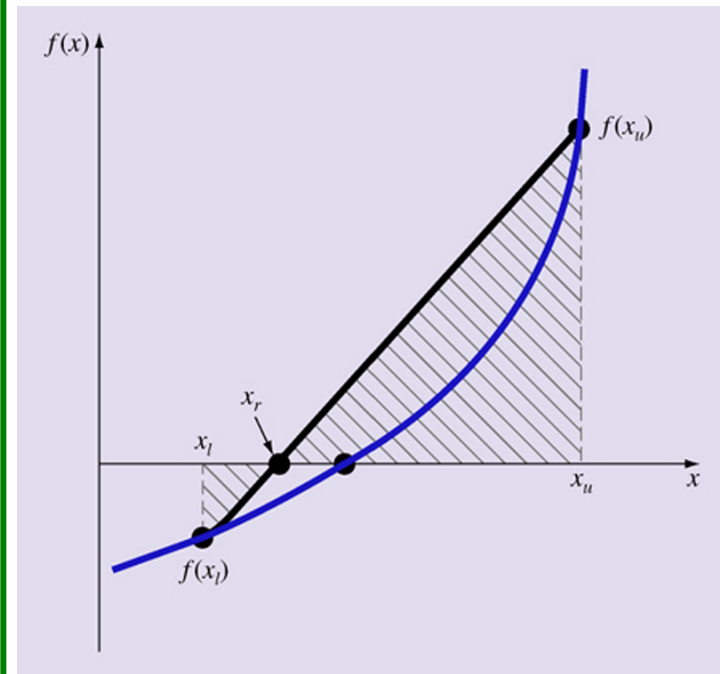
서로 곱하면  $f(x_l)(x_r - x_u) = f(x_u)(x_r - x_l)$

정리하여

$$x_r[f(x_l) - f(x_u)] = x_u f(x_l) - x_l f(x_u)$$

$f(x_l) - f(x_u)$  로 나누면

$$x_r = \frac{x_u f(x_l)}{f(x_l) - f(x_u)} - \frac{x_l f(x_u)}{f(x_l) - f(x_u)}$$



## 5.5 가위치법 (3/3)



### 가위치법 공식의 유도

$x_u$  를 더하고 빼면

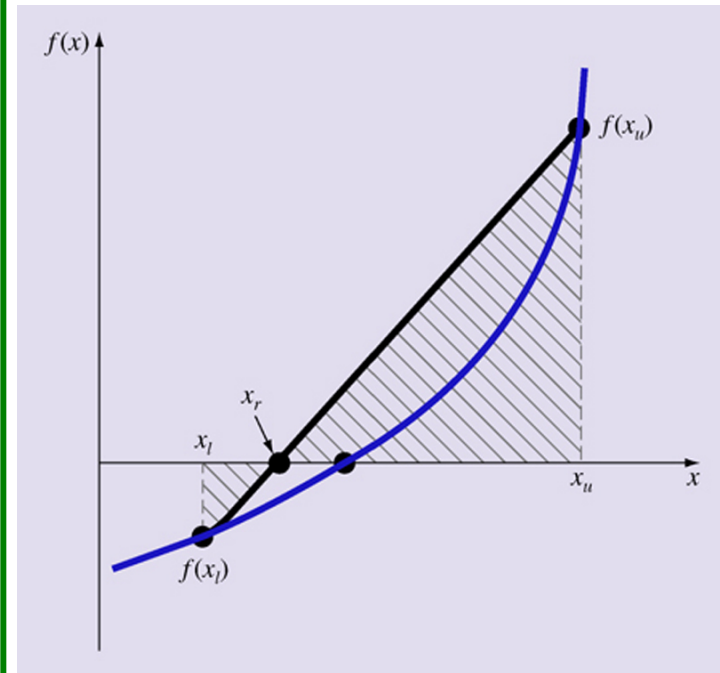
$$x_r = x_u + \frac{x_u f(x_l)}{f(x_l) - f(x_u)} - x_u - \frac{x_l f(x_u)}{f(x_l) - f(x_u)}$$

항을 모으면

$$x_r = x_u + \frac{x_u f(x_l)}{f(x_l) - f(x_u)} - \frac{x_l f(x_u)}{f(x_l) - f(x_u)}$$

즉

$$x_r = x_u - \frac{f(x_u)(x_l - x_u)}{f(x_l) - f(x_u)}$$



## 예제 5.5 [1/2]

Q. 가위치법을 사용하여 예제 5.1의 방정식의 근을 구하라.

풀이)

- 첫 번째 반복에 의해

$$x_l = 50 \quad f(x_l) = -4.579387$$

$$x_u = 200 \quad f(x_u) = 0.860291$$

$$x_r = 200 - \frac{0.860291(50 - 200)}{-4.579387 - 0.860291} = 176.2773$$

$$\varepsilon_t = 23.5\%$$



## 예제 5.5 [2/2]

- 두 번째 반복에 의해

$$f(x_l)f(x_r) = -2.592732$$

$$x_l = 50 \qquad f(x_l) = -4.579387$$

$$x_u = 176.2773 \qquad f(x_u) = 0.860291$$

$$x_r = 176.2773 - \frac{0.566174(50 - 176.2773)}{-4.579387 - 0.566174} = 162.3828$$

$$\varepsilon_t = 13.76\%, \quad \varepsilon_a = 8.56\%$$



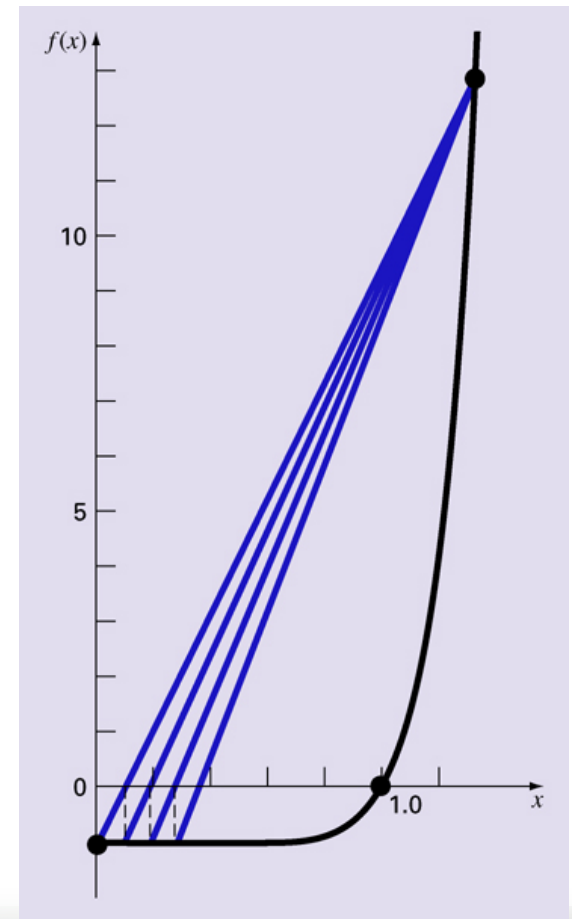
## 예제 5.6(이분법이 가위치법 보다 바람직한 경우) [1/3]

Q. 이분법과 가위치법을 사용해서  $x = 0$ 과  $1.3$  사이에서  $f(x) = x^{10} - 1$  의 근을 구하라.

풀이)

- 이분법

반복	$x_l$	$x_u$	$x_r$	$\epsilon_a$ (%)	$\epsilon_t$ (%)
1	0	1.3	0.65	100.0	35.0
2	0.65	1.3	0.975	33.3	2.5
3	0.975	1.3	1.1375	14.3	13.8
4	0.975	1.1375	1.05625	7.7	5.6
5	0.975	1.05625	1.015625	4.0	1.6

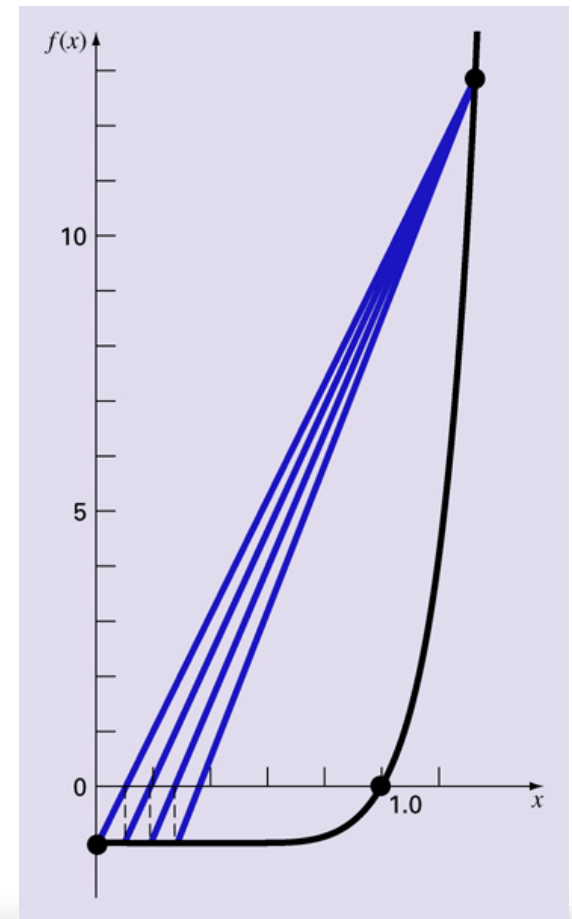




# 예제 5.6(이분법이 가위치법 보다 바람직한 경우) [2/3]

## • 가위치법

반복	$x_l$	$x_u$	$x_r$	$\epsilon_a$ (%)	$\epsilon_t$ (%)
1	0	1.3	0.09430		90.6
2	0.09430	1.3	0.18176	48.1	81.8
3	0.18176	1.3	0.26287	30.9	73.7
4	0.26287	1.3	0.33811	22.3	66.2
5	0.33811	1.3	0.40788	17.1	59.2



## 예제 5.6(이분법이 가위치법 보다 바람직한 경우) (3/3)

- 가위치법이 이분법보다 항상 우수한 것은 아니다.
- 계산결과가 빨리 수렴되는지를 검사할 필요가 있다.
- 구간의 끝점 중의 하나가 고정되어 한쪽 방향으로만 수렴한다. → 느린 수렴속도

