جــامـــعـــة دمـــشـــق
كـليـة الهـنـدسـة المـدنيـة

تـصـمـيــم بـمـعـونــة الـحــاســب
Computer Aided Design

Lecture 6 (last one)

Numerical Solvers

16/12/2025

Wael Darwich

Damascus University

Civil Engineering Faculty

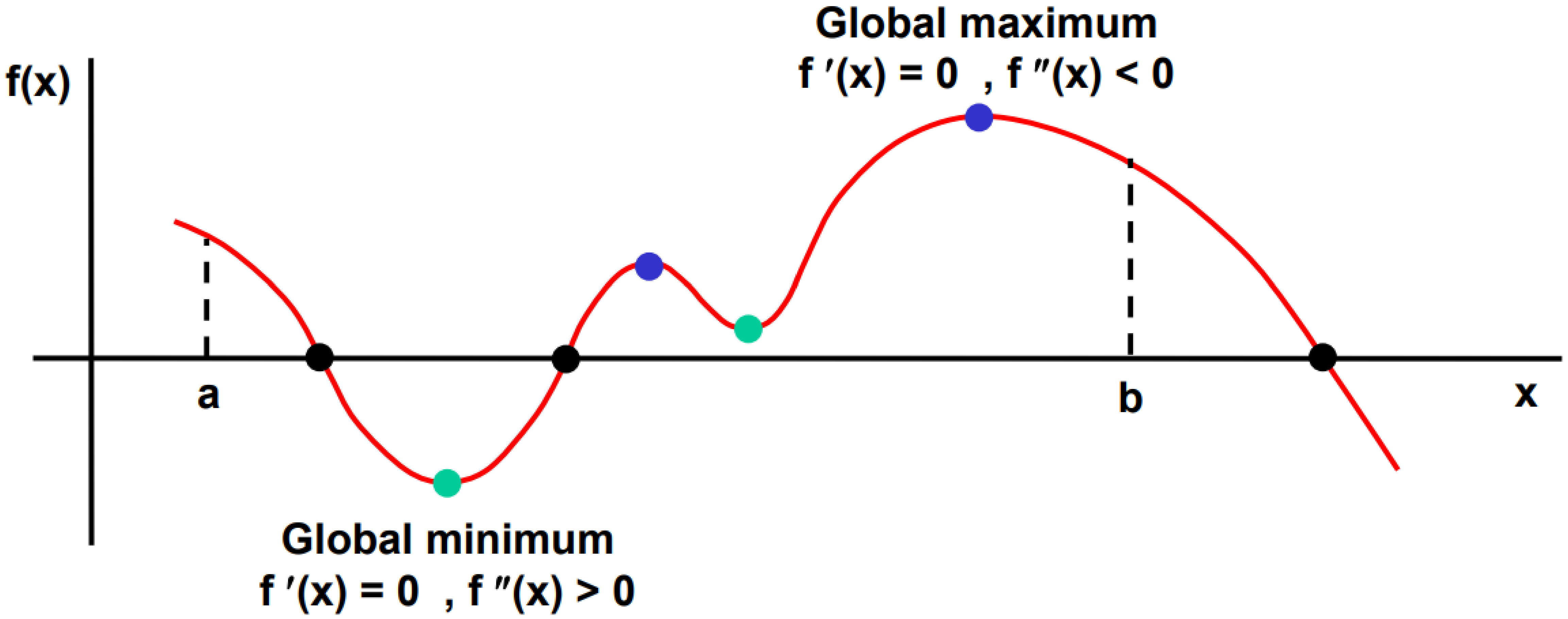
Last Lecture

- Mathematical Programming
  - Linear Programming
- Optimization Examples

Content

- Numerical Solvers
  - Bisection
  - Newton–Raphson
  - Golden–Section Search
- Example
- Exam

Numerical Solvers



Bisection method

- Root finding tool
- Continuous functions
- Start with initial values of opposite signs
- Root position inside initial range
- Reduce range
- Get root within numerical tolerance

Bisection

Find root of:

$$f(x) = 0.15x^3 - x^2 + x + 10$$

```
y = @(x)0.15*x.^3-x.^2+x+10;
fzero(y, -5)
-2.3720
```

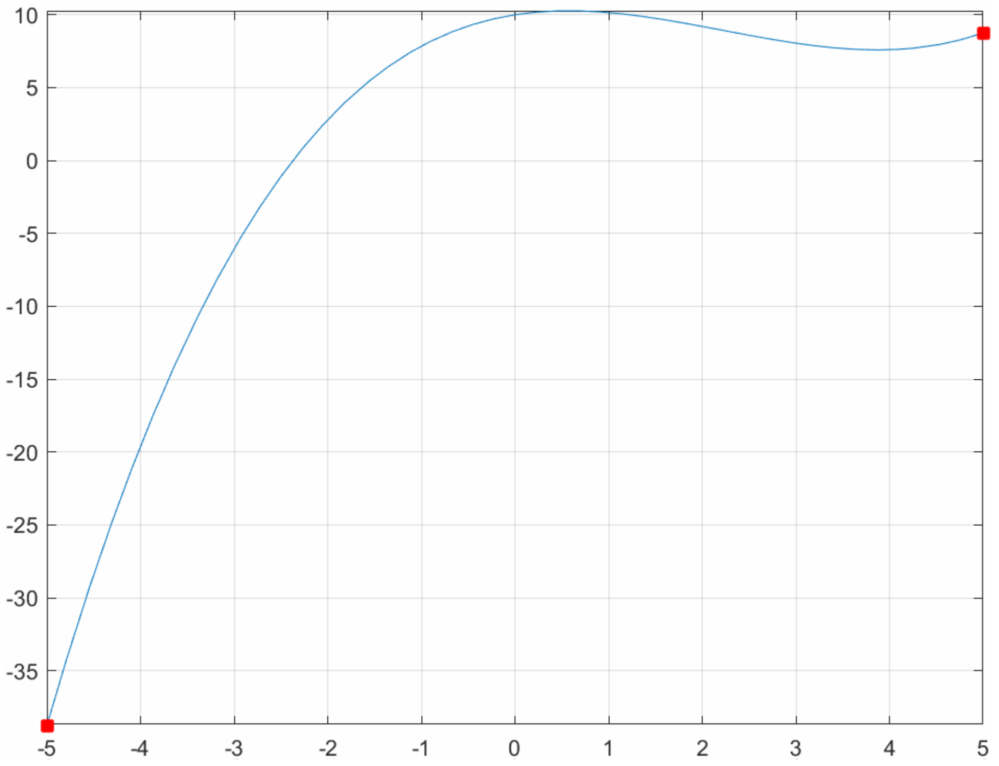
Bisection

Find root of:

$$f(x) = 0.15x^3 - x^2 + x + 10$$

```
y = @(x)0.15*x.^3-x.^2+x+10;
x1 = -5; x2 = 5;

step = 0.1;
for i = 1:100
        y1 = y(x1);
        y2 = y(x2);
        plot([x1 x2], [y1 y2], 'r+')
        if (y1 < 0)
                x1 = x1 + step;
        end
        if (y2 > 0)
                x2 = x2 - step;
        end
end
```



Bisection

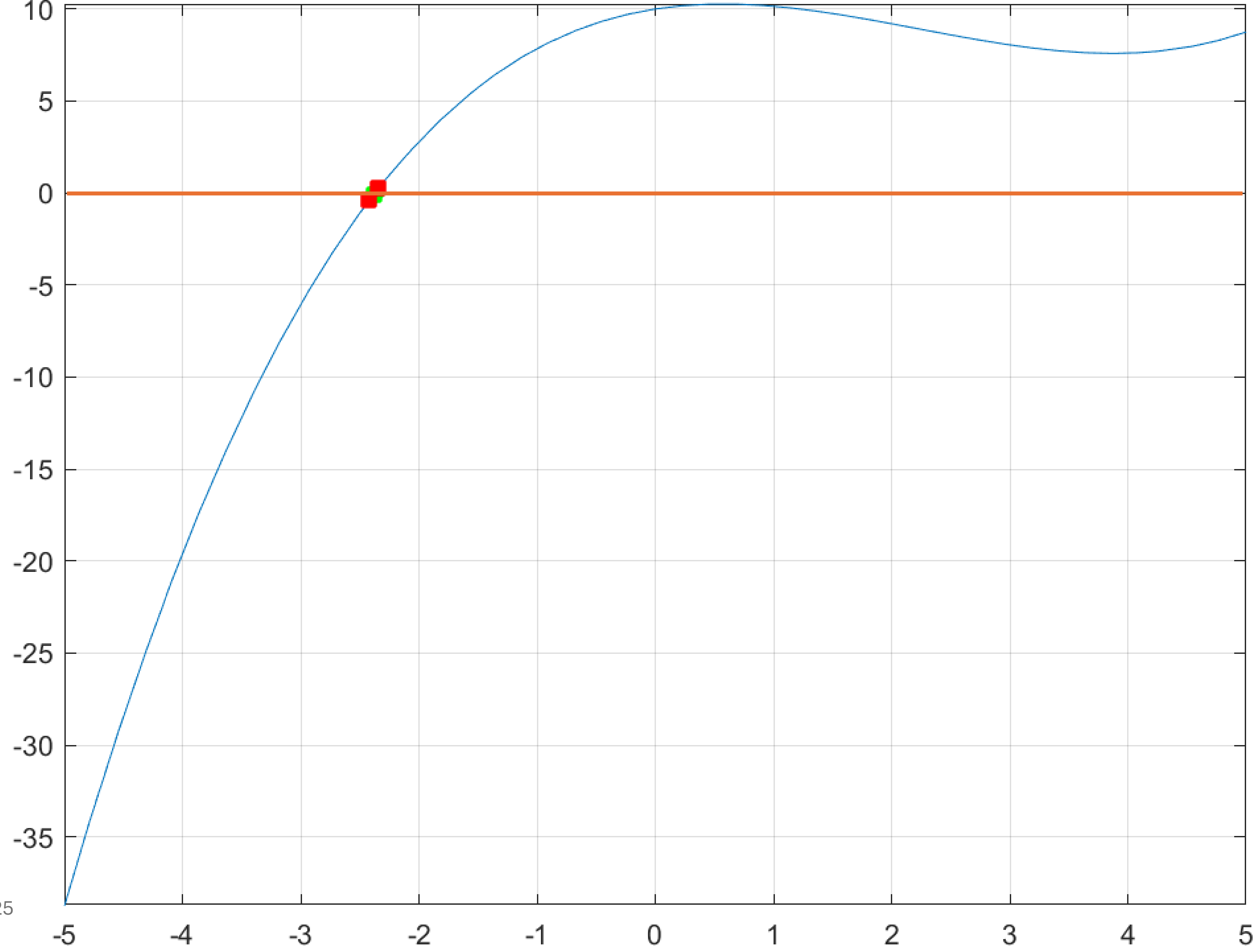
Itr.	x1	x2	y
1	-5.00	5.00	-15.00000
2	-4.90	4.90	-14.01000
3	-4.80	4.80	-13.04000
4	-4.70	4.70	-12.09000
5	-4.60	4.60	-11.16000
6	-4.50	4.50	-10.25000
…			
67	-2.30	-1.60	2.90527
68	-2.30	-1.70	2.62900
69	-2.30	-1.80	2.33507
70	-2.30	-1.90	2.02305
71	-2.30	-2.00	1.69247
72	-2.30	-2.10	1.34290
73	-2.30	-2.20	0.97387
74	-2.30	-2.30	0.58495
75	-2.30	-2.40	0.17567

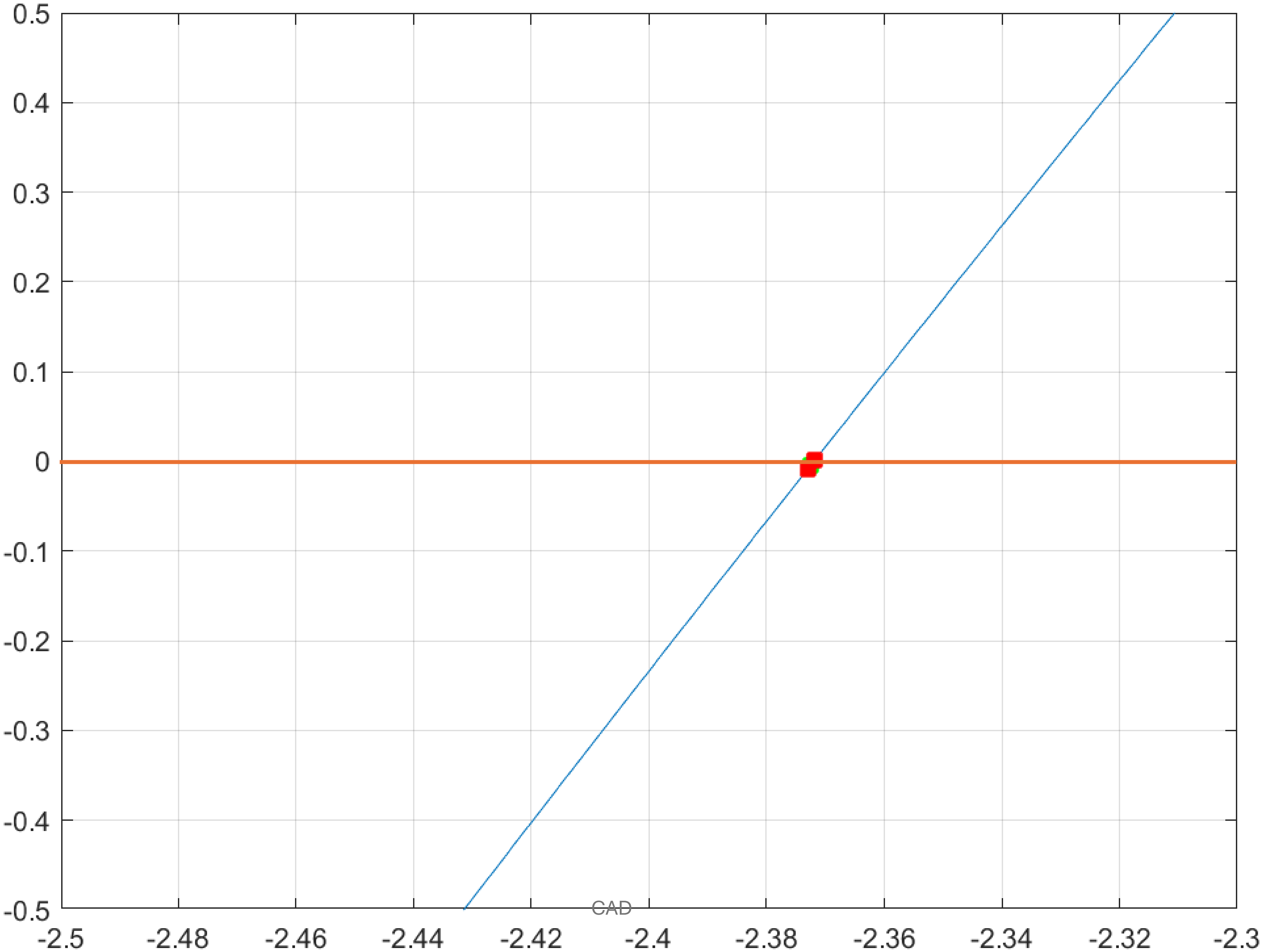
Bisection

Itr.	x1	x2	y
1	-5.00	5.00	-15.00000
2	-4.90	4.90	-14.01000
3	-4.80	4.80	-13.04000
4	-4.70	4.70	-12.09000
5	-4.60	4.60	-11.16000
6	-4.50	4.50	-10.25000
…			
67	-2.30	-1.60	2.90527
68	-2.30	-1.70	2.62900
69	-2.30	-1.80	2.33507
70	-2.30	-1.90	2.02305
71	-2.30	-2.00	1.69247
72	-2.30	-2.10	1.34290
73	-2.30	-2.20	0.97387
74	-2.30	-2.30	0.58495
75	**-2.30**	**-2.40**	0.17567

Bisection

- Number of steps
- Step size, fixed vs variable
- Bisection: split half range each iteration
  - Keep range end points on different y signs


10
5
0
-5
-10
-15
-20
-25
-30
-35
-5
-4
-3
-2
-1
0
1
2
3
4
5



0.5
0.4
0.3
0.2
0.1
0
-0.1
-0.2
-0.3
-0.4
-0.5
-2.5
-2.48
-2.46
-2.44
-2.42
-2.4
-2.38
-2.36
-2.34
-2.32
-2.3

Bisection

```
function [x3, x4, ym] = bisec(x1, x2, y)
    y1 = y(x1);     y2 = y(x2);     xm = (x1+x2)/2;     ym = y(xm);

    if y1*y2 > 0
        disp('Cannot have both end points at the same side')
        return
    end

    if y1*ym > 0
        x3 = xm; x4 = x2;
    else
        x3 = x1; x4 = xm;
    end

end
```

Bisection

```
y = @(x)0.15*x.^3-x.^2+x+10;

x1 = -5;
x2 = 5;


ym = 1;

while abs(ym) > 1e-4
    [x1, x2, ym] = bisec(x1, x2, y);

end
```

Bisection

Itr.	x1	x2	y
1	-5.00	5.00	10.00000
2	-5.00	0.00	-1.09375
3	-2.50	0.00	6.89453
4	-2.50	-1.25	3.62061
5	-2.50	-1.88	1.45721
6	-2.50	-2.19	0.23190
7	-2.50	-2.34	-0.41817
8	-2.42	-2.34	-0.08998
9	-2.38	-2.34	0.07175
10	-2.38	-2.36	-0.00892
11	-2.37	-2.36	0.03146
12	-2.37	-2.37	0.01129
13	-2.37	-2.37	0.00119
14	-2.37	-2.37	-0.00386
15	-2.37	-2.37	-0.00134
16	-2.37	-2.37	-0.00008

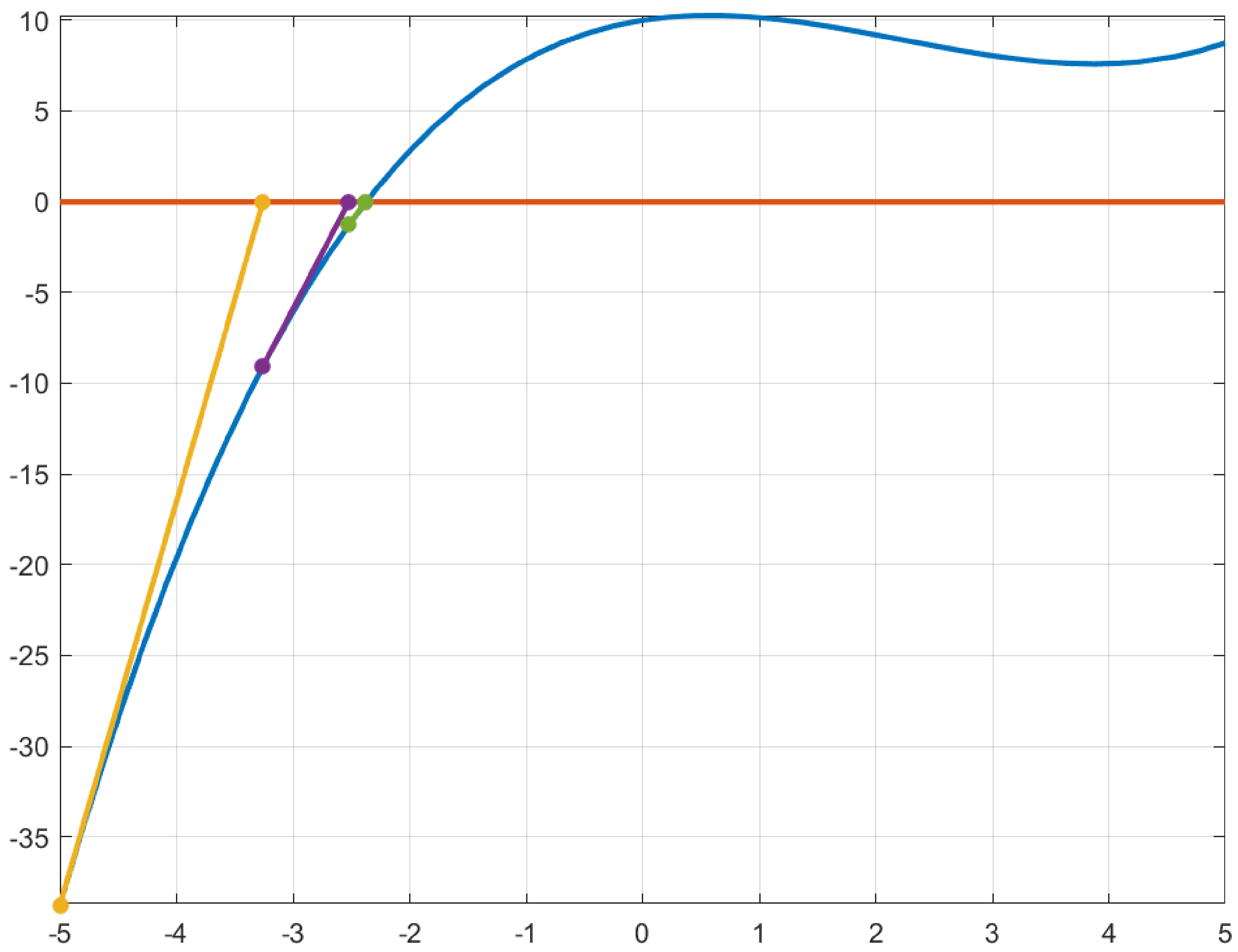
Bisection

Itr.	x1	x2	y
1	-5.00	5.00	10.00000
2	-5.00	0.00	-1.09375
3	-2.50	0.00	6.89453
4	-2.50	-1.25	3.62061
5	-2.50	-1.88	1.45721
6	-2.50	-2.19	0.23190
7	-2.50	-2.34	-0.41817
8	-2.42	-2.34	-0.08998
9	-2.38	-2.34	0.07175
10	-2.38	-2.36	-0.00892
11	-2.37	-2.36	0.03146
12	-2.37	-2.37	0.01129
13	-2.37	-2.37	0.00119
14	-2.37	-2.37	-0.00386
15	-2.37	-2.37	-0.00134
16	**-2.37**	**-2.37**	-0.00008

Itr.	x1	x2	y
1	-5.00	5.00	-15.00000
2	-4.90	4.90	-14.01000
3	-4.80	4.80	-13.04000
4	-4.70	4.70	-12.09000
5	-4.60	4.60	-11.16000
6	-4.50	4.50	-10.25000
…			
67	-2.30	-1.60	2.90527
68	-2.30	-1.70	2.62900
69	-2.30	-1.80	2.33507
70	-2.30	-1.90	2.02305
71	-2.30	-2.00	1.69247
72	-2.30	-2.10	1.34290
73	-2.30	-2.20	0.97387
74	-2.30	-2.30	0.58495
75	**-2.30**	**-2.40**	0.17567

Newton–Raphson

- Root finding tool
- Continuous differentiable function
- Start with initial guess, e.g. $x=-5$
- Iterate for better solution
  - linear approximation



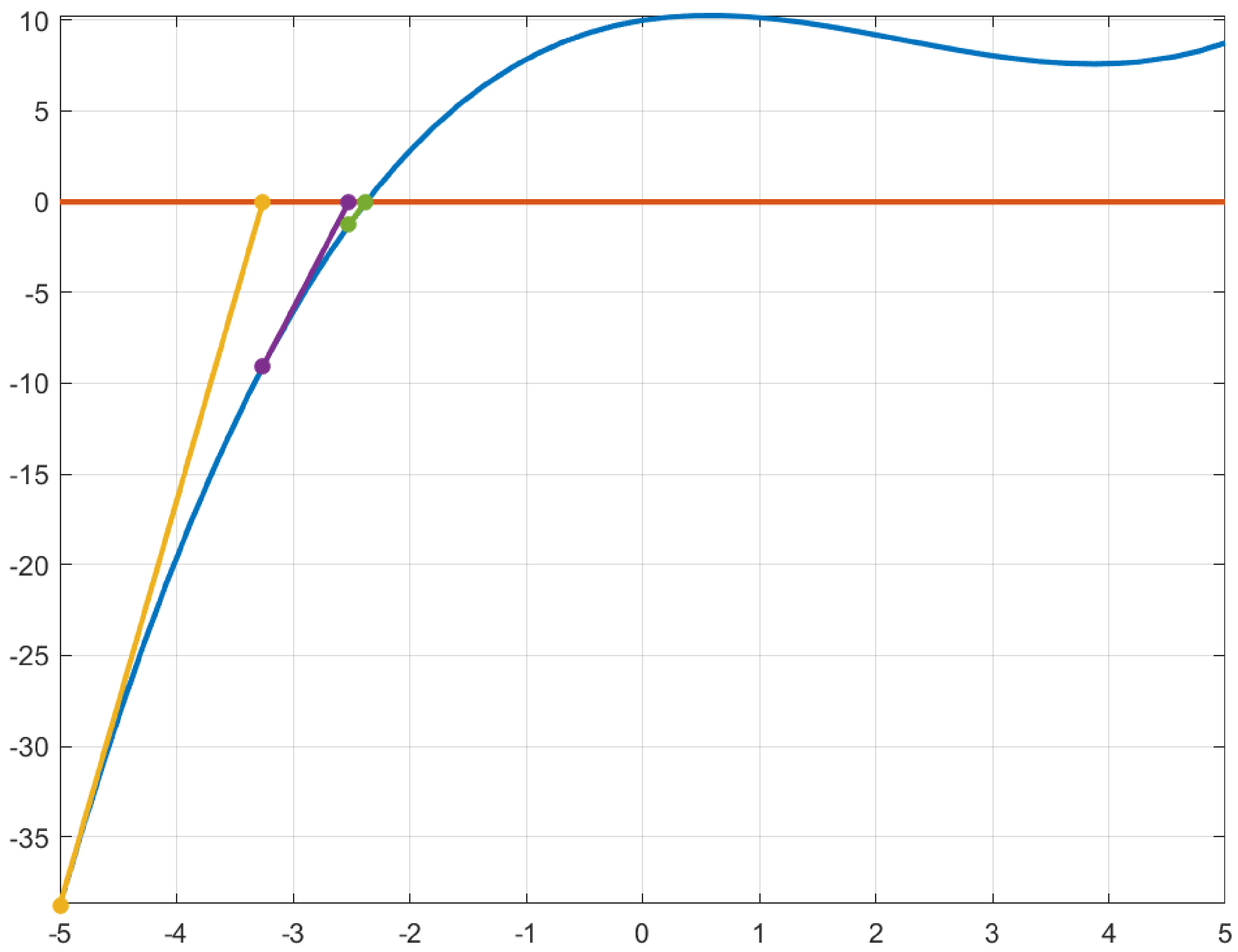
Newton–Raphson

$$f'(x_n) = \frac{\Delta y}{\Delta x}$$

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

$$f'(x_n) = \frac{0 - f(x_n)}{x_{n+1} - x_n}$$

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

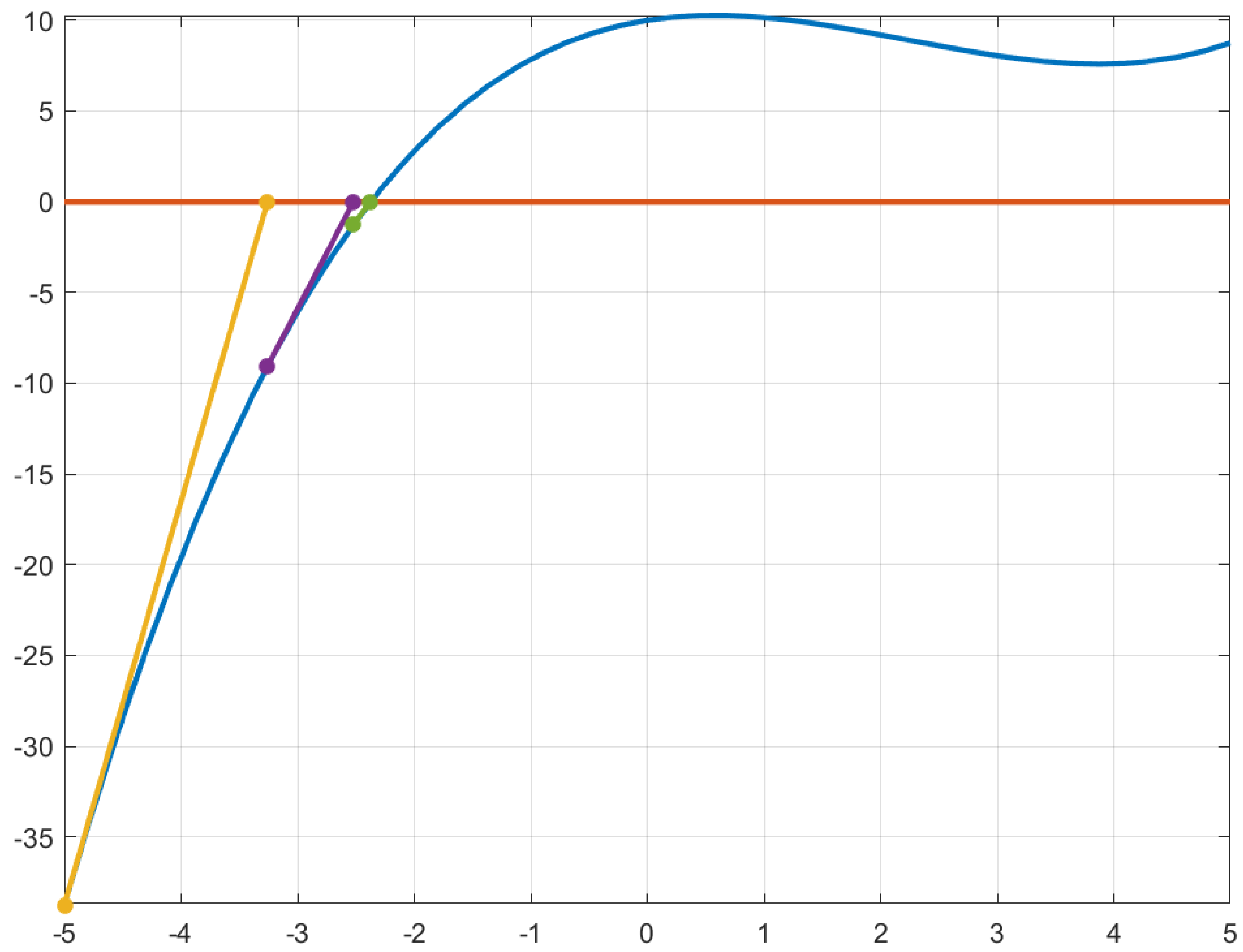


Newton–Raphson

```
syms x
y = 0.15*x.^3-x.^2+x+10;
dy = diff(y,x);

x1 = -5;
fplot(y)

y1 = 1;
while abs(y1) > 1e-4
        y1 = subs(y, x1);
        dy1 = subs(dy, x1);
        x2 = x1 - y1/dy1;
        plot([x1 x2], [y1 0], '-*')
        x1 = x2;
end
```



Newton–Raphson

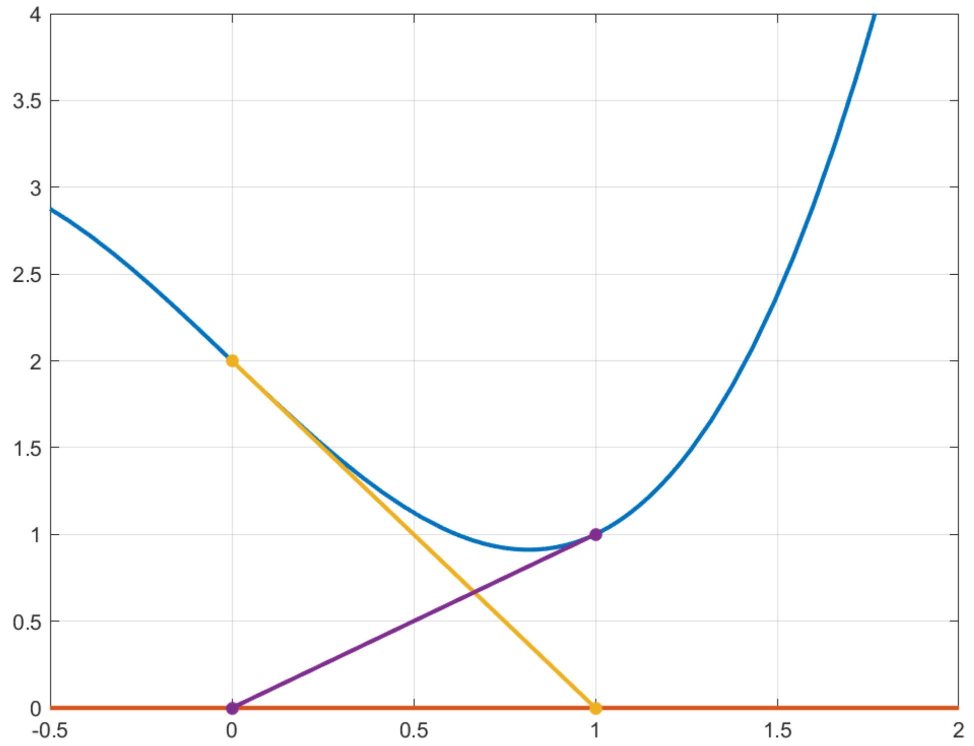
- Needs function derivative
  - no vertical tangent line
  - can get numerical slope using two points
  - secant method: slower, easier
- Derivative cannot be zero

$$f'(x) = 0$$

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

- Oscillating forever

$$x^3 - 2x + 2$$



Newton–Raphson vs. Bisection

Find roots methods:

	Bisection	Newton-Raphson
Convergence Rate	Slow linear	Faster quadratic
Convergence Guarantee	Guaranteed if there is a root within initial range	Convergence is not guaranteed and is sensitive to the initial guess
Initial Guess	Requires two initial guesses in opposite signs	Requires a single initial guess that must be close to the actual root
Function Requirements	a continuous function	a differentiable function
Computational Cost/Iter	Low (one function evaluation)	Higher (function and derivative evaluation)
Complexity	Simpler to implement	More complex due to derivative requirement

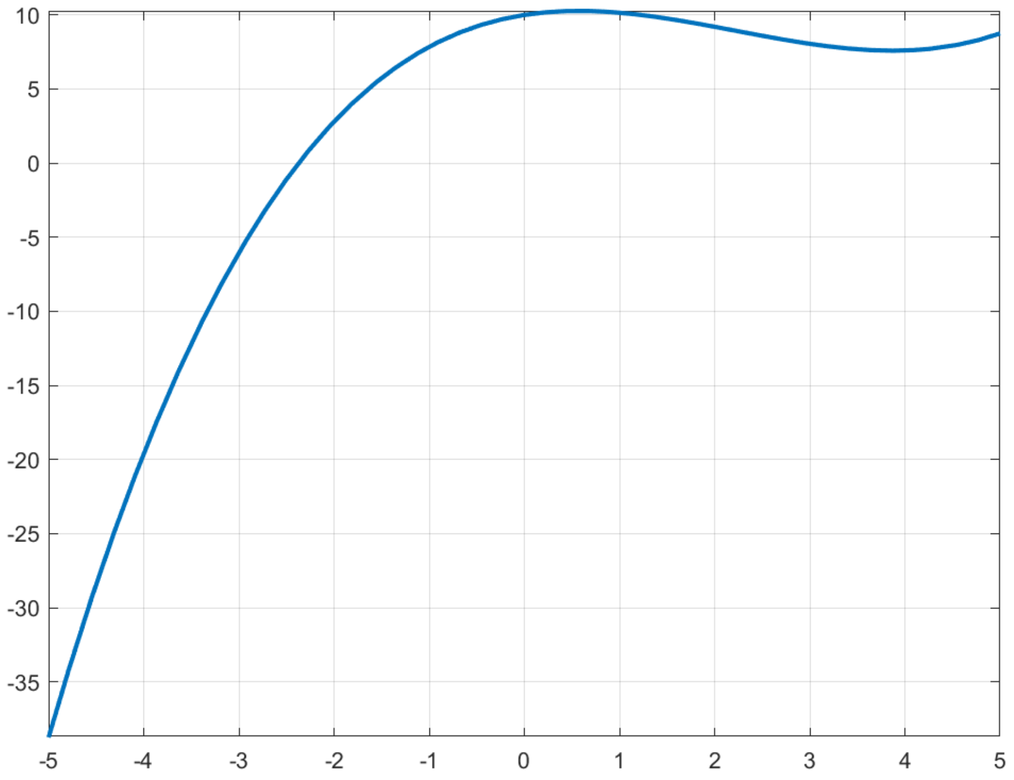
Local Maximum

Find maximum value of:

$$f(x) = 0.15x^3 - x^2 + x + 10$$

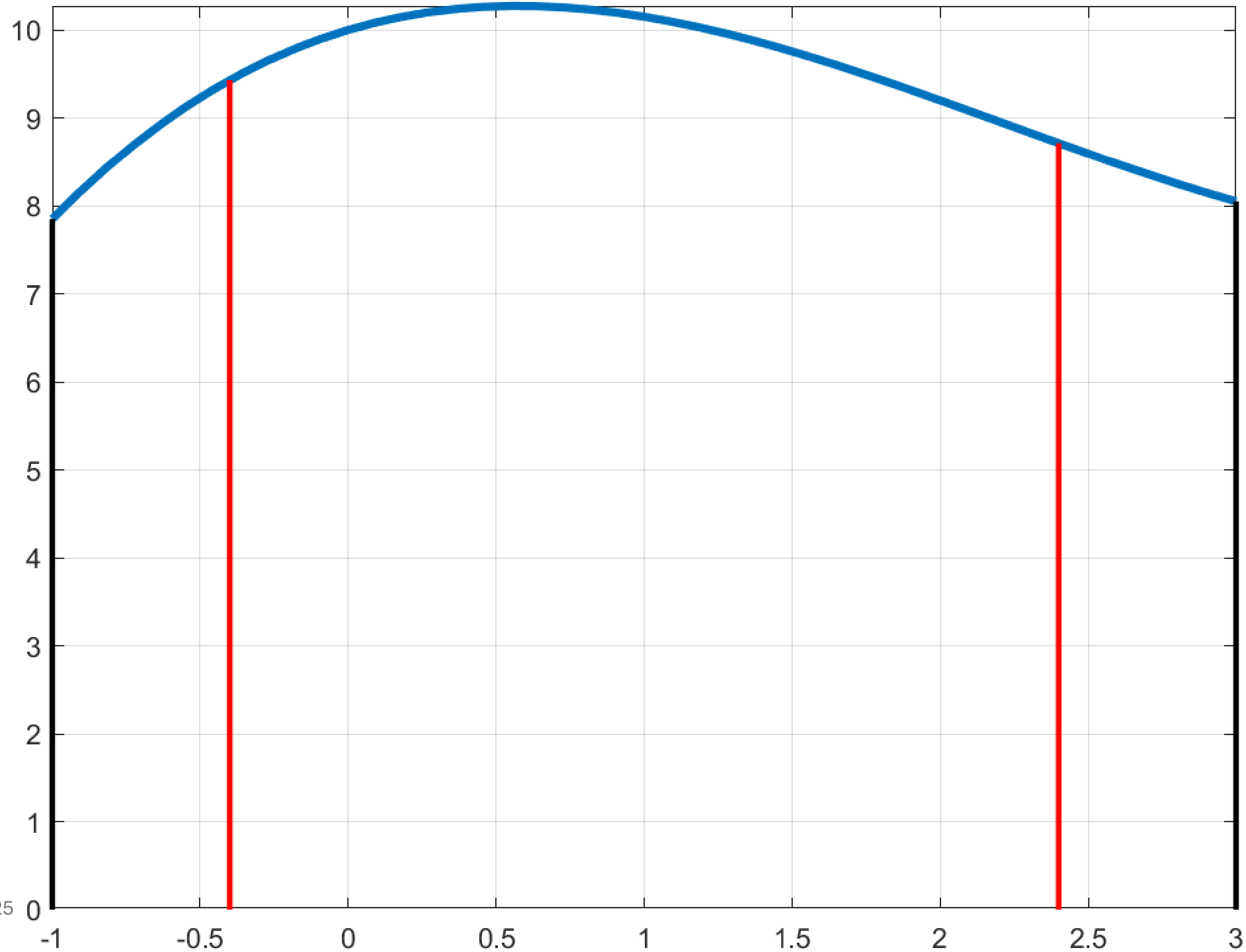
In the range `[-5 5]`

```
y = @(x)   0.15*x.^3-x.^2+x+10;
fplot(y, [-5 5])
y2= @(x) -(0.15*x.^3-x.^2+x+10);
x = fminbnd(y2, -5, 5) % 0.5742
y(x)                   % 10.2729
```



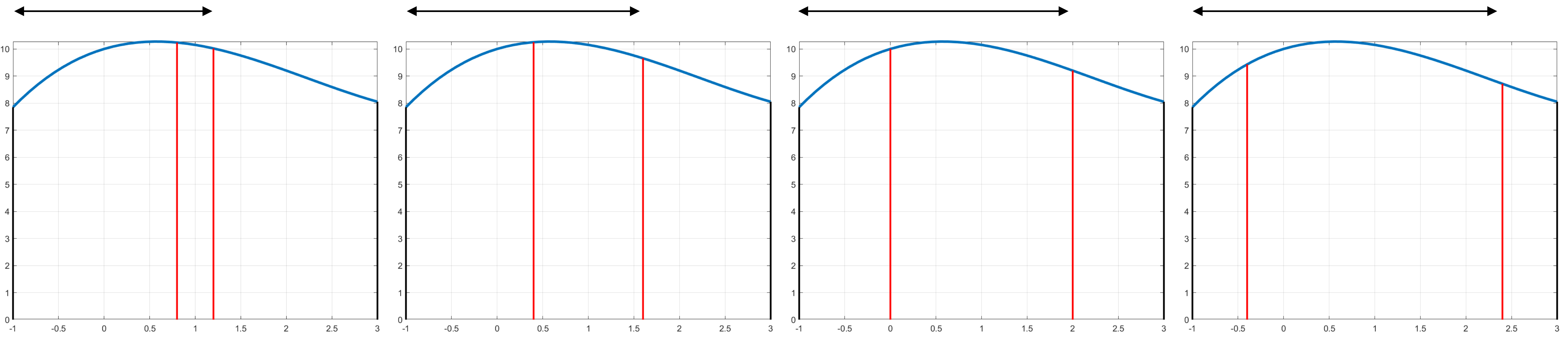
Golden–Section Search

- Extremum finding tool
- In a given range
- Only for unimodal function
  - Single solution in range
- Reduce range size
  - y is unknown (unlike root)

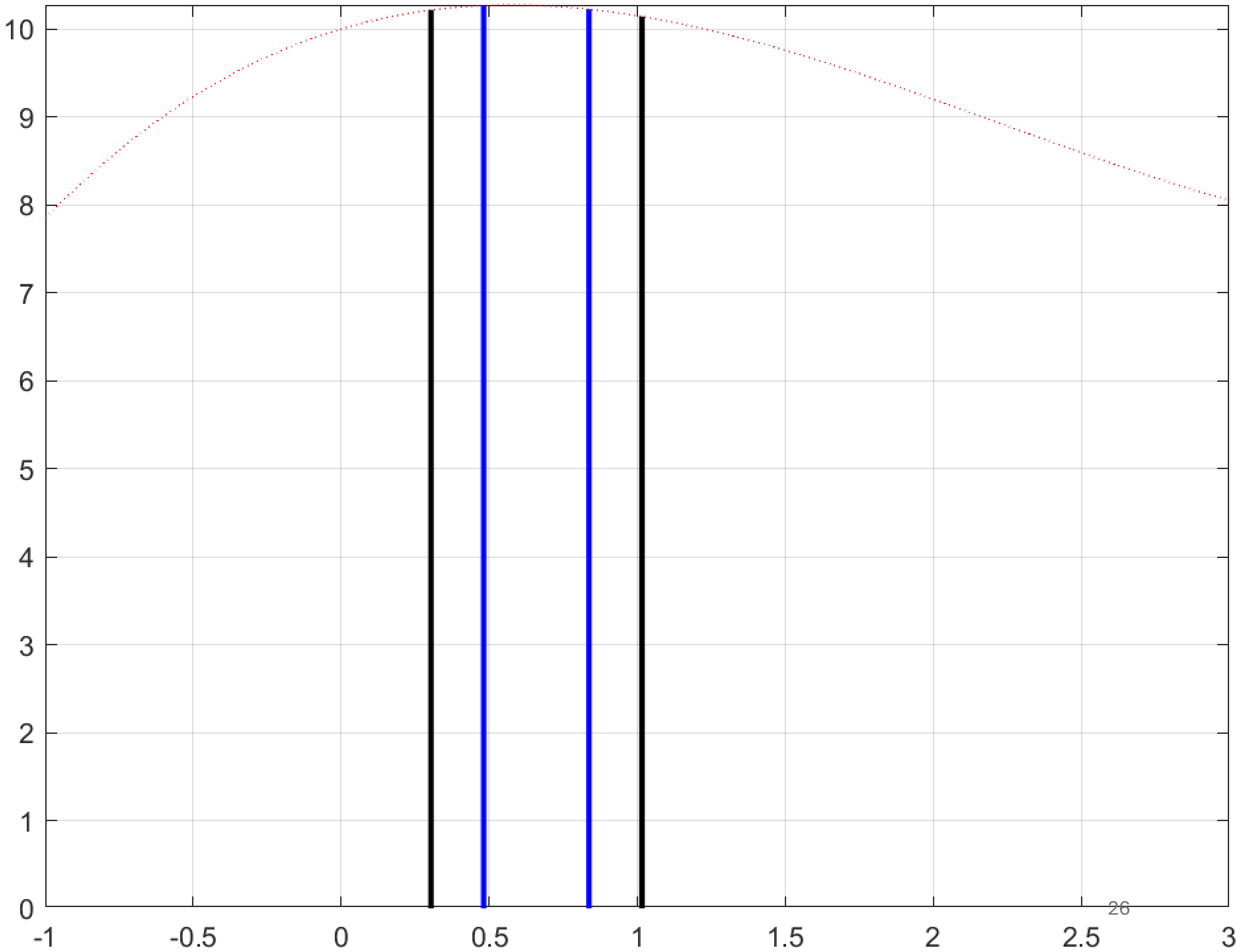

10
9
8
7
6
5
4
3
2
1
0
-1
-0.5
0
0.5
1
1.5
2
2.5
3

Golden–Section Search

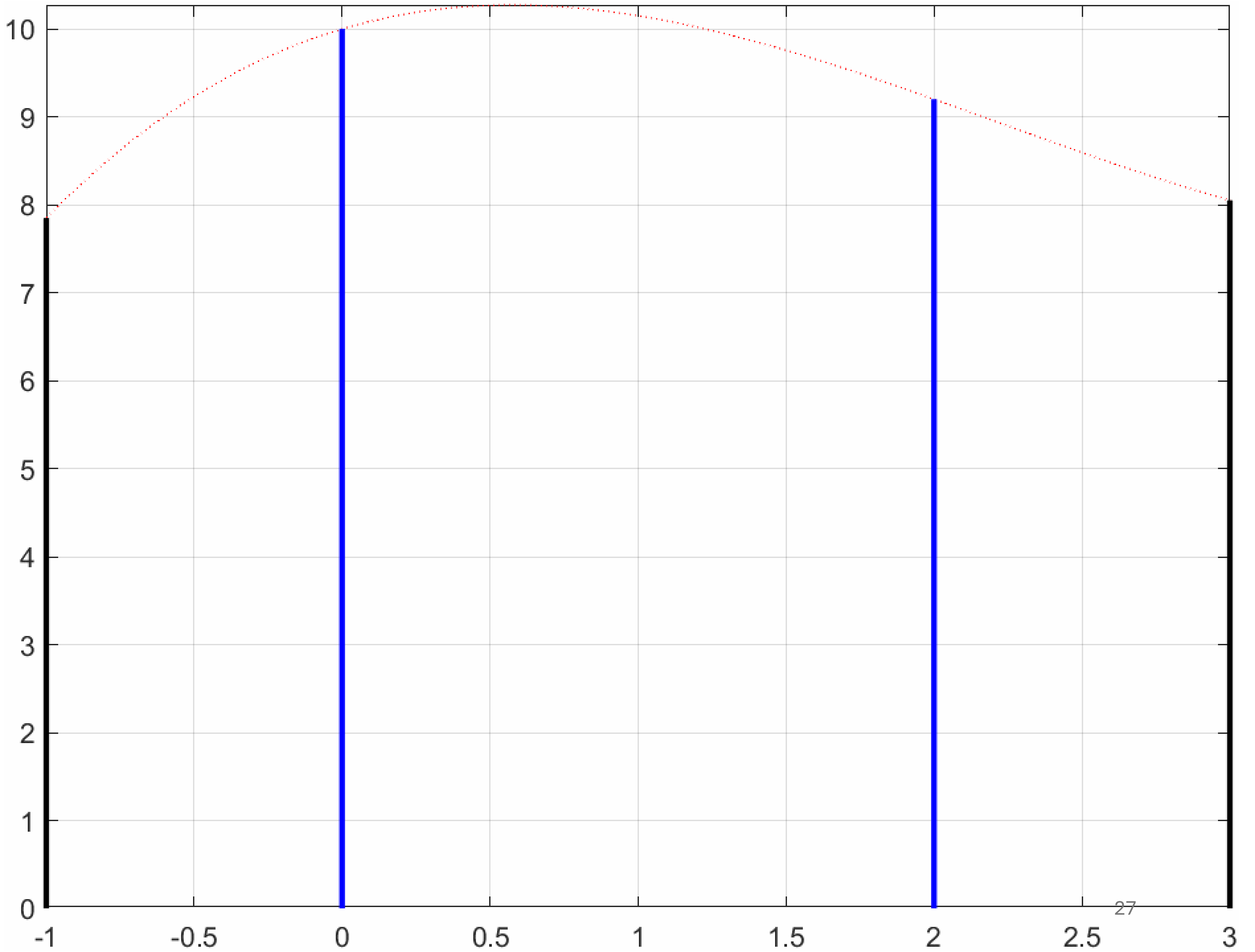
- $y_1 > y_2 \Rightarrow$ maximum on the left



-1.0000	3.0000
-1.0000	2.0000
-0.2500	2.0000
-0.2500	1.4375
-0.2500	1.0156
0.0664	1.0156
0.3037	1.0156
0.3037	0.8376
0.4372	0.8376
0.4372	0.7375
0.4372	0.6624
0.4935	0.6624
0.4935	0.6202
0.5252	0.6202
0.5489	0.6202
0.5489	0.6024
0.5489	0.5890
0.5590	0.5890

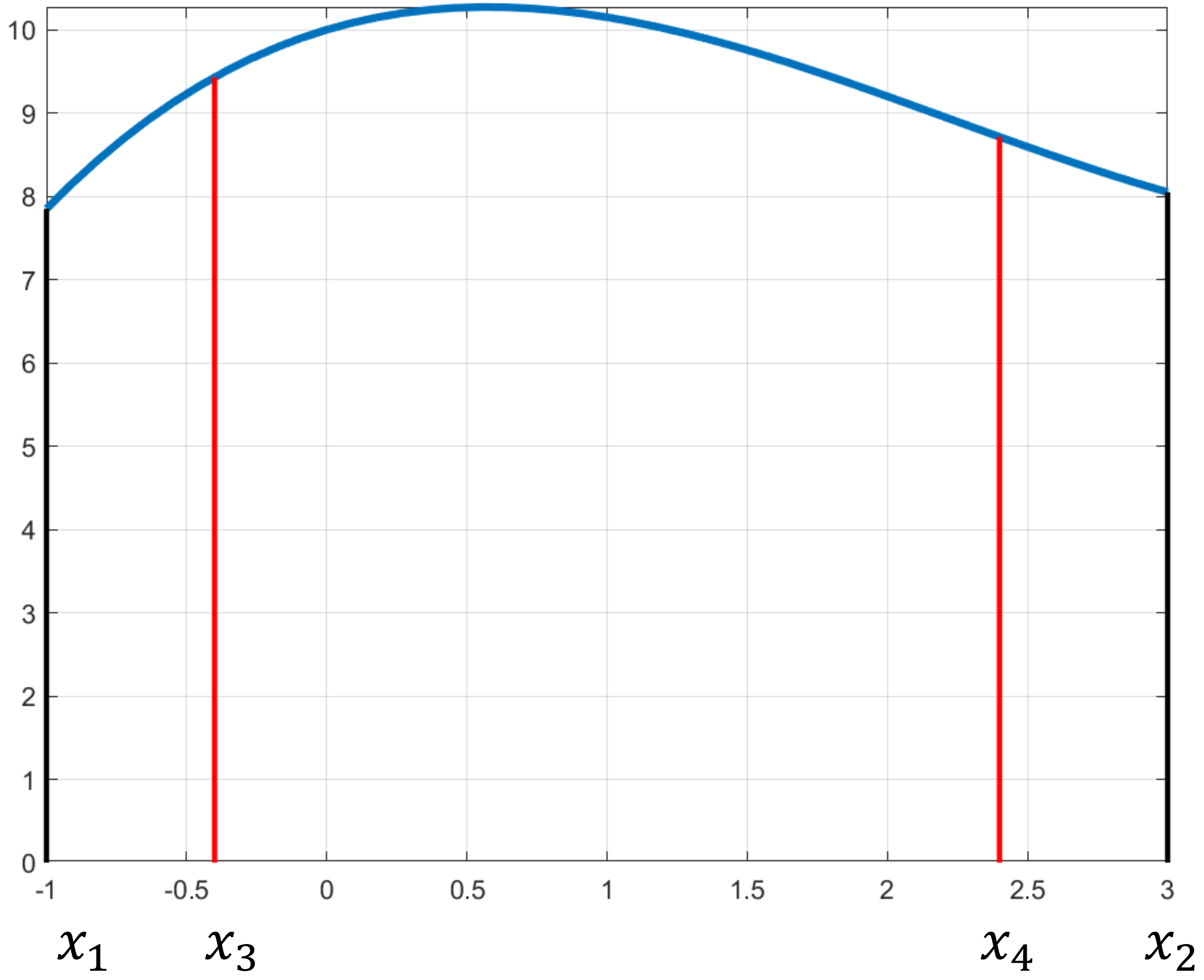


```
    x   =  0.5742
y(x)  = 10.2729
```



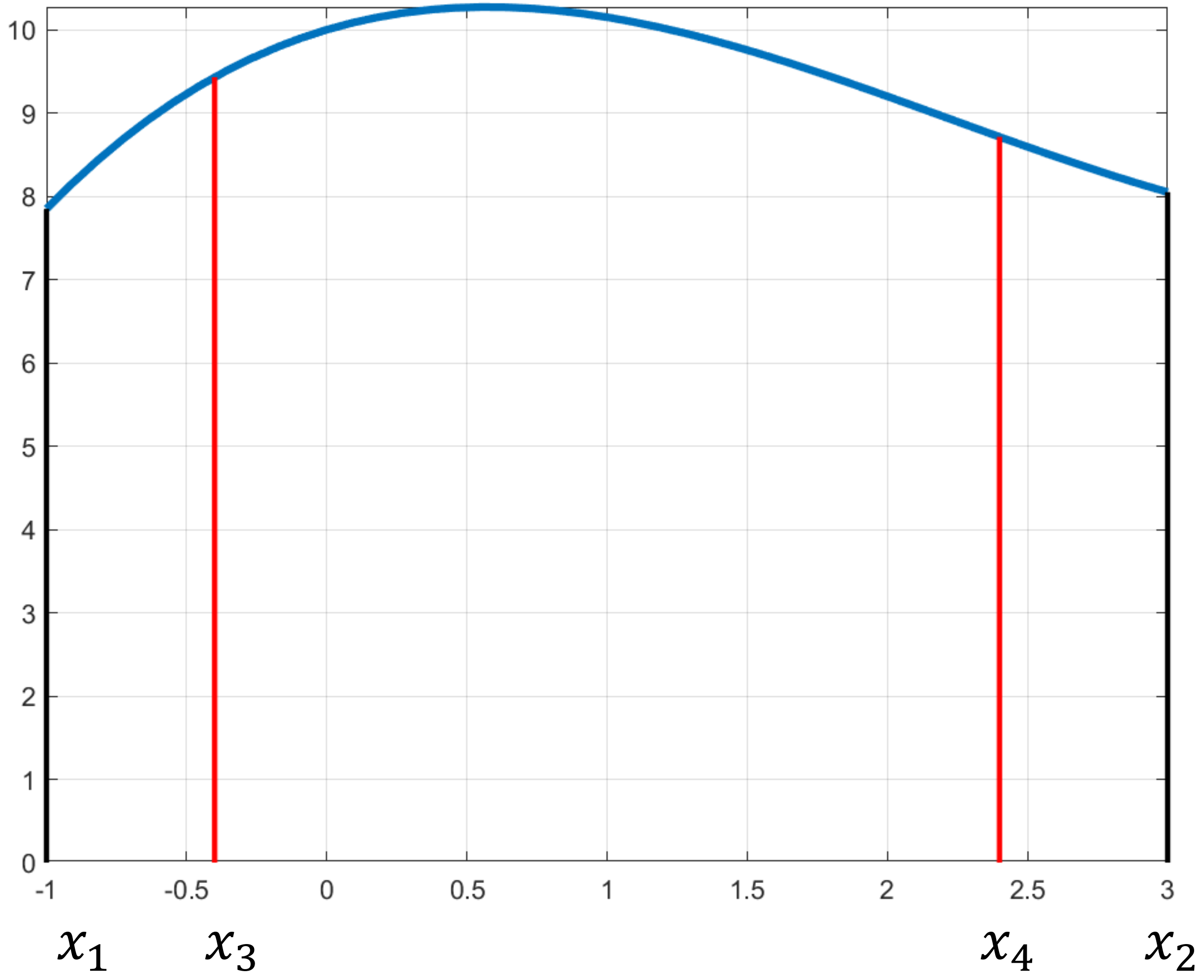
Golden–Section Search

```
y = @(x)0.15*x.^3-x.^2+x+10;
x1 = -1; x2 = 3;
ratio = 0.25;
epsilon = 1e-4;
while abs(x2-x1) > epsilon
    d = x2-x1;
    x3 = x1 + ratio * d;
    x4 = x2 - ratio * d;
    y3 = y(x3);
    y4 = y(x4);
    if (y3 > y4)
        x2 = x4;
    else
        x1 = x3;
    end
end
```

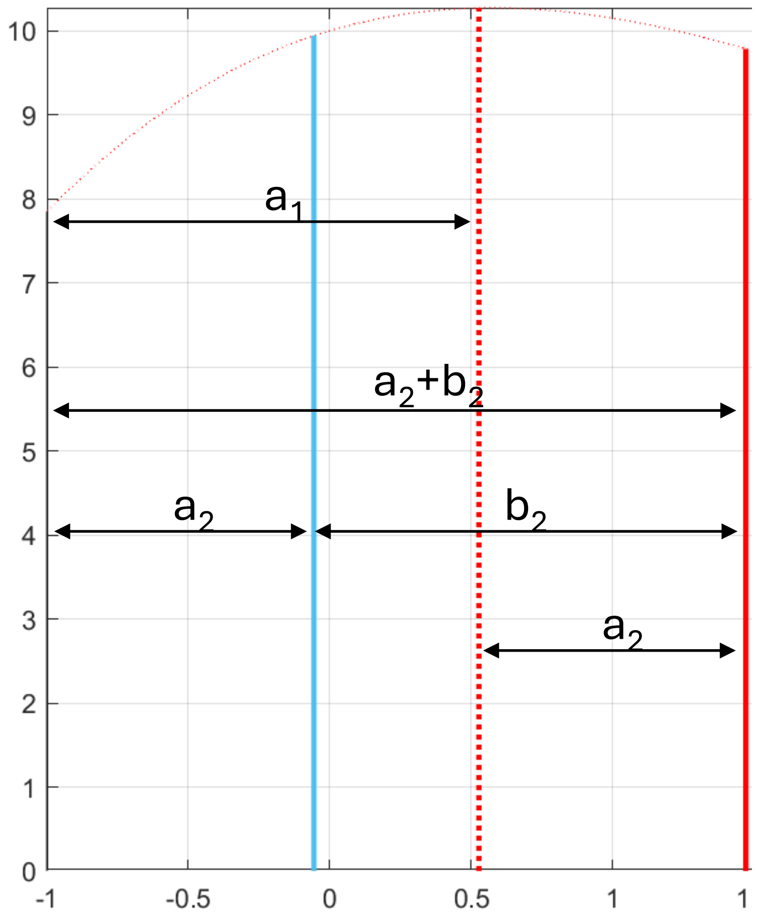


Golden–Section Search

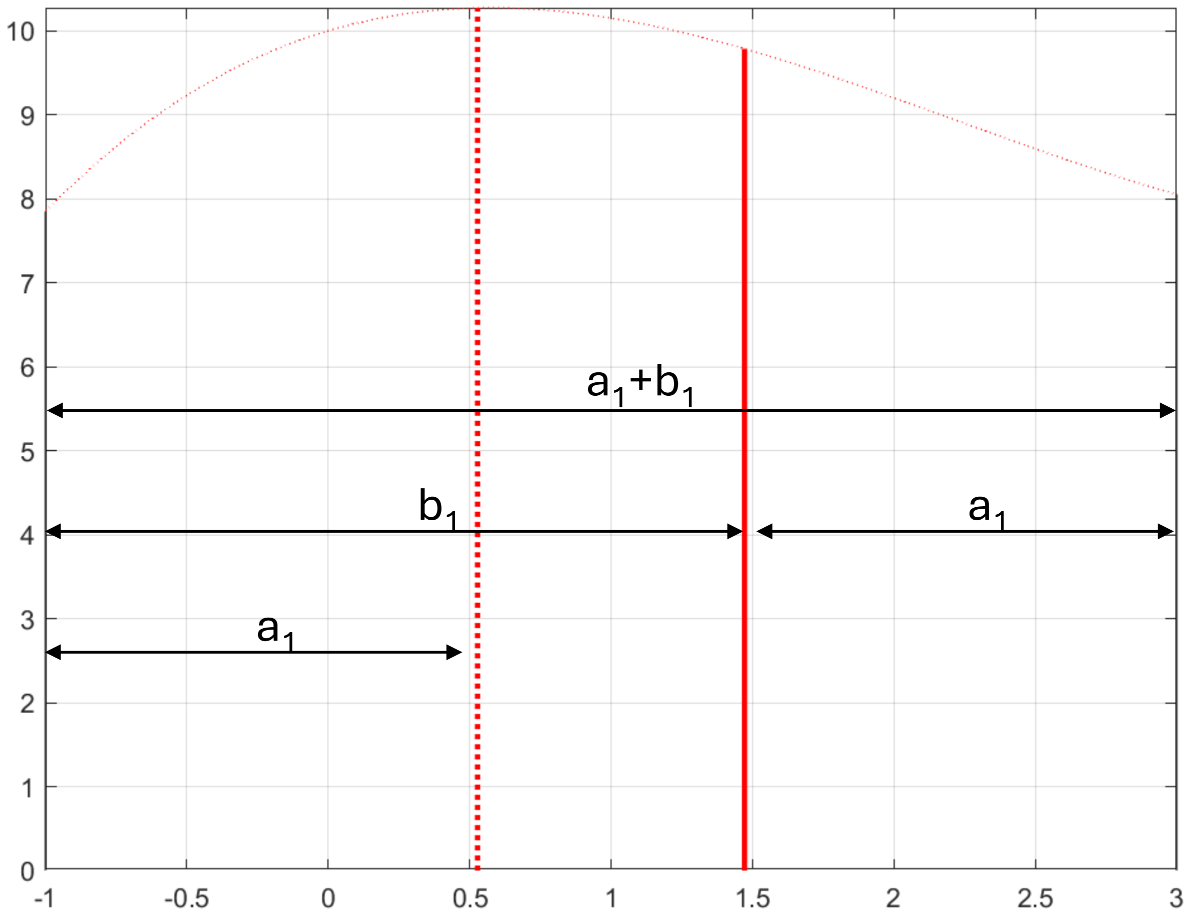
```
y = @(x)0.15*x.^3-x.^2+x+10;
x1 = -1; x2 = 3;
ratio = 0.25;
epsilon = 1e-4;
while abs(x2-x1) > epsilon
    d = x2-x1;
    x3 = x1 + ratio * d;
    x4 = x2 - ratio * d;
    y3 = y(x3);
    y4 = y(x4);
    if (y3 > y4)
        x2 = x4;
    else
        x1 = x3;
    end
end
```



Golden–Section Search



Iteration 2



Iteration 1

Golden ratio when: $b_2 = a_1$

Golden–Section Search

Ration in first iteration

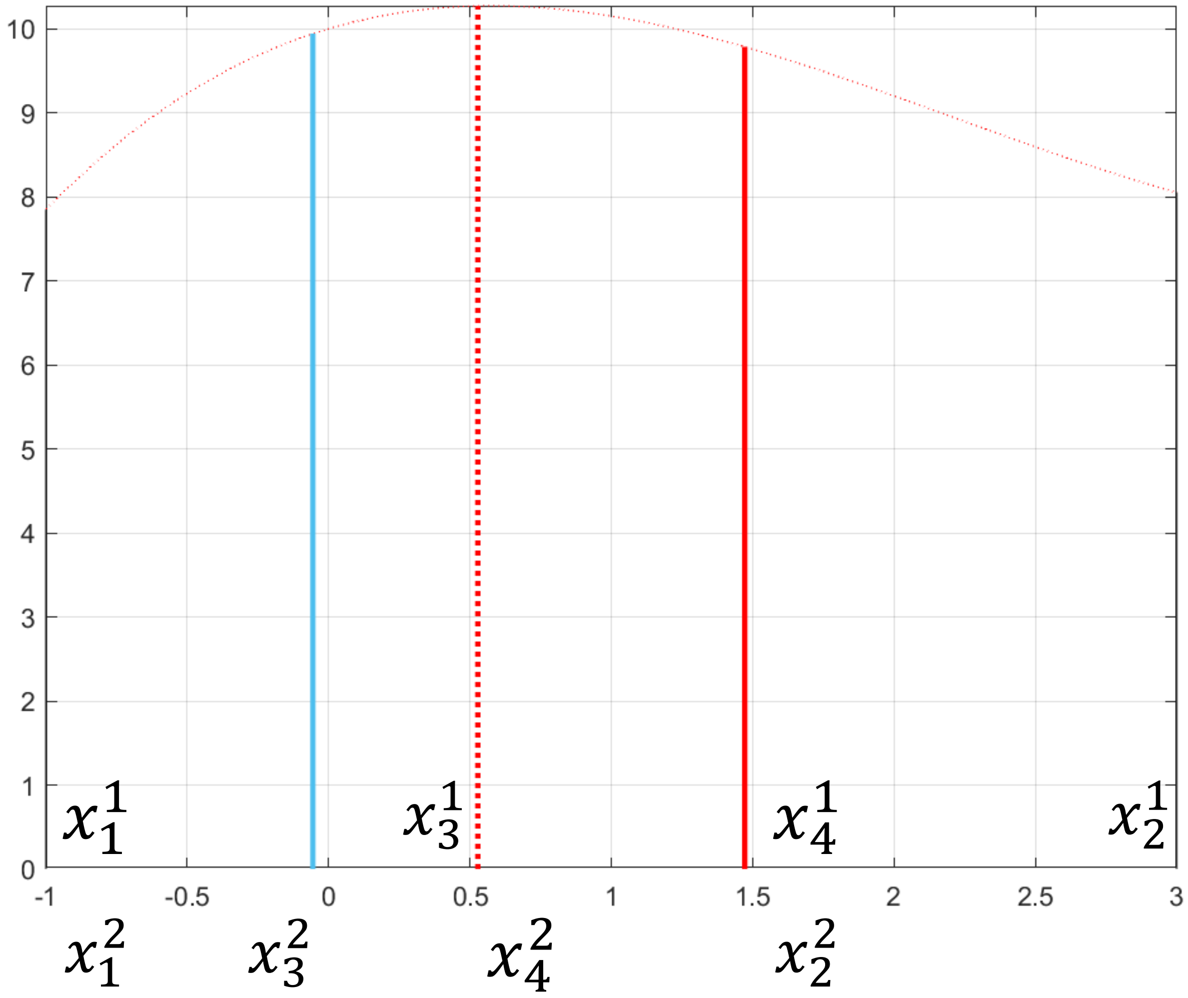
$$\frac{x_4^1 - x_1^1}{x_2^1 - x_1^1}$$

Ration in second iteration

$$\frac{x_2^2 - x_3^2}{x_2^2 - x_1^2} = \frac{x_2^2 - x_3^2}{x_4^1 - x_1^1}$$

Golden ratio when:

$$x_2^2 - x_3^2 = x_2^1 - x_4^1$$



Golden–Section Search

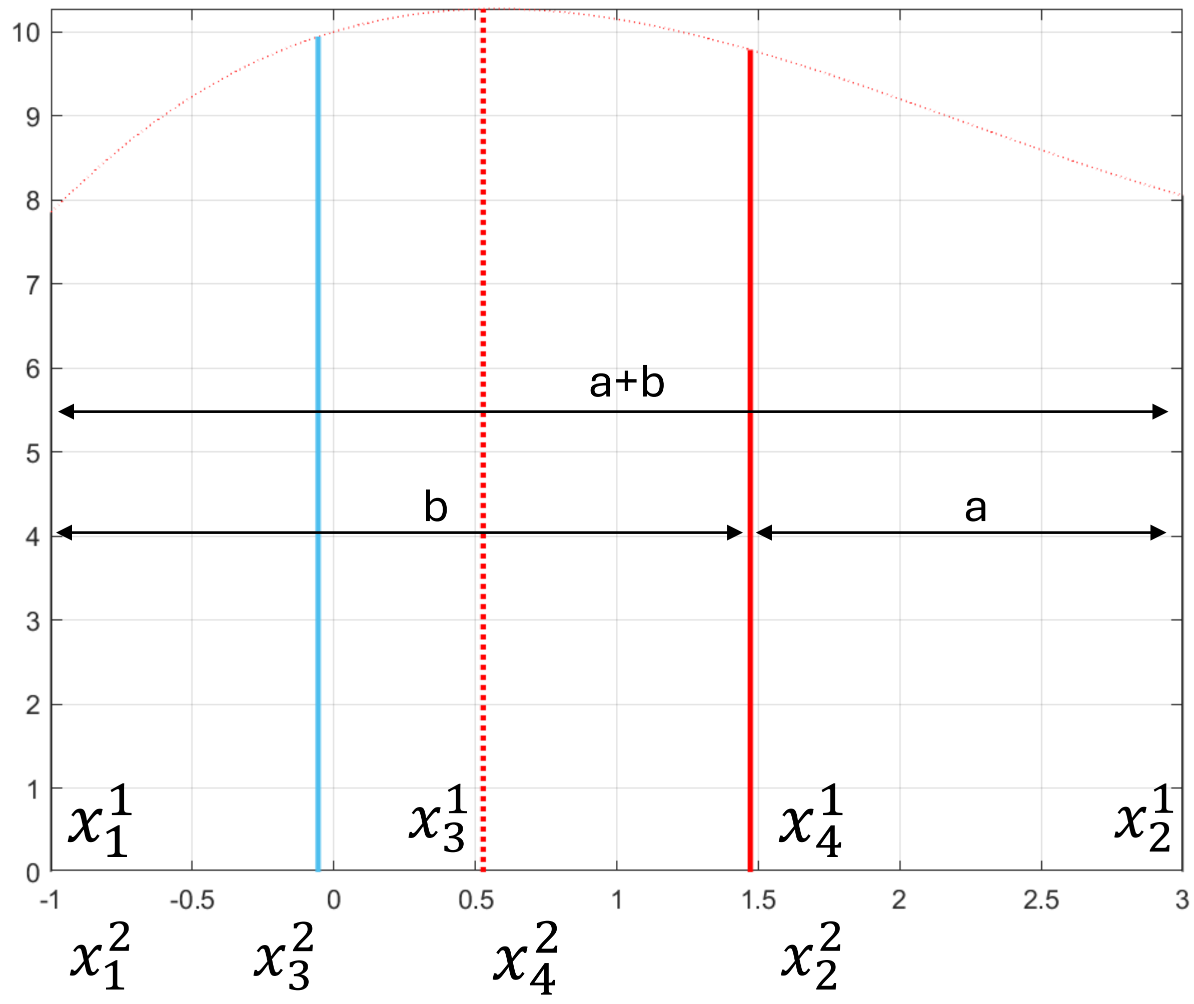
Same ratio

$$\frac{x_4^1 - x_1^1}{x_2^1 - x_1^1} = \frac{x_2^1 - x_4^1}{x_4^1 - x_1^1}$$

$$\frac{b}{a+b} = \frac{a}{b} = Ratio$$

$$\frac{a+b}{b} = \frac{b}{a}$$

$$R + 1 = \frac{1}{R}$$



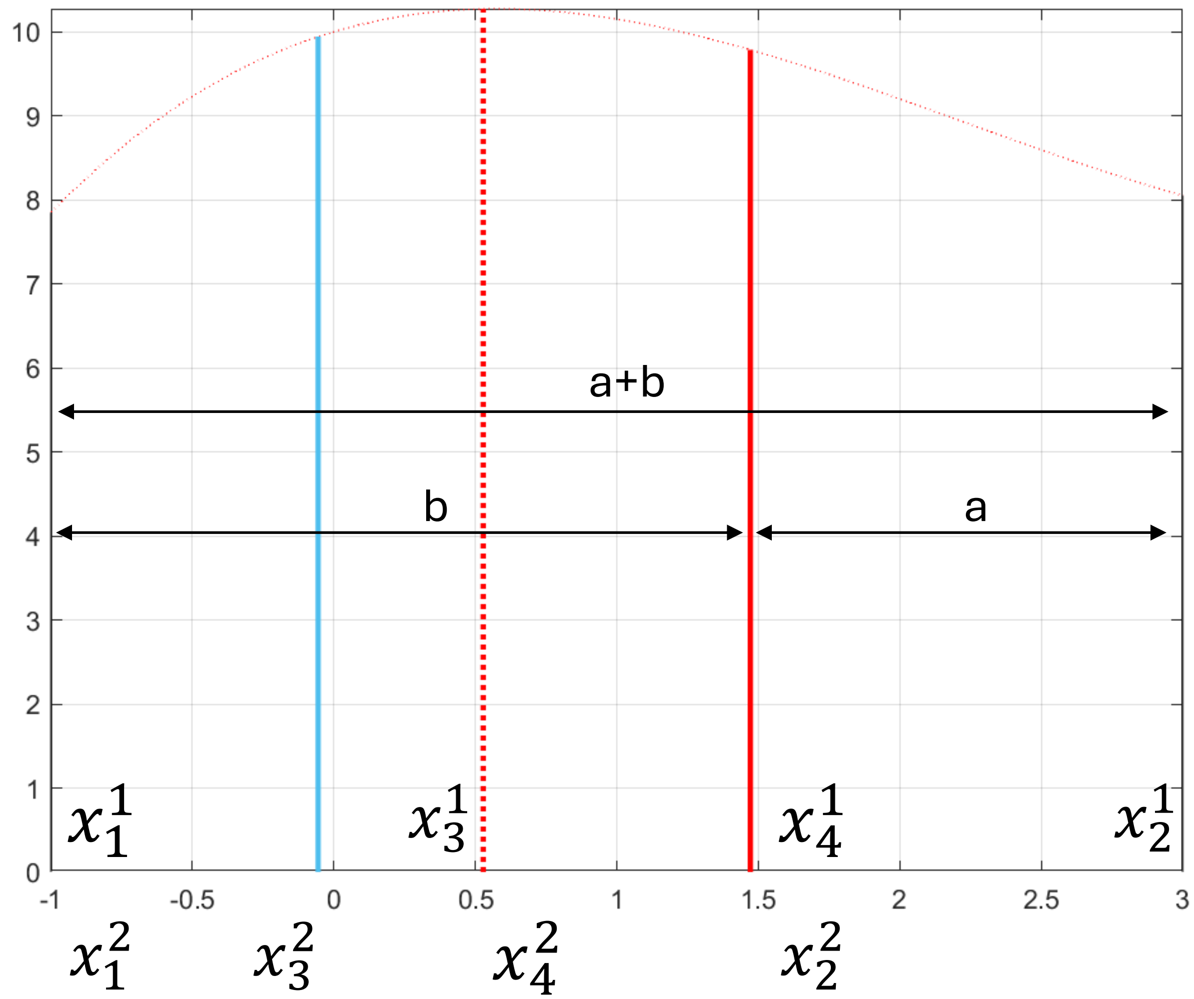
Golden–Section Search

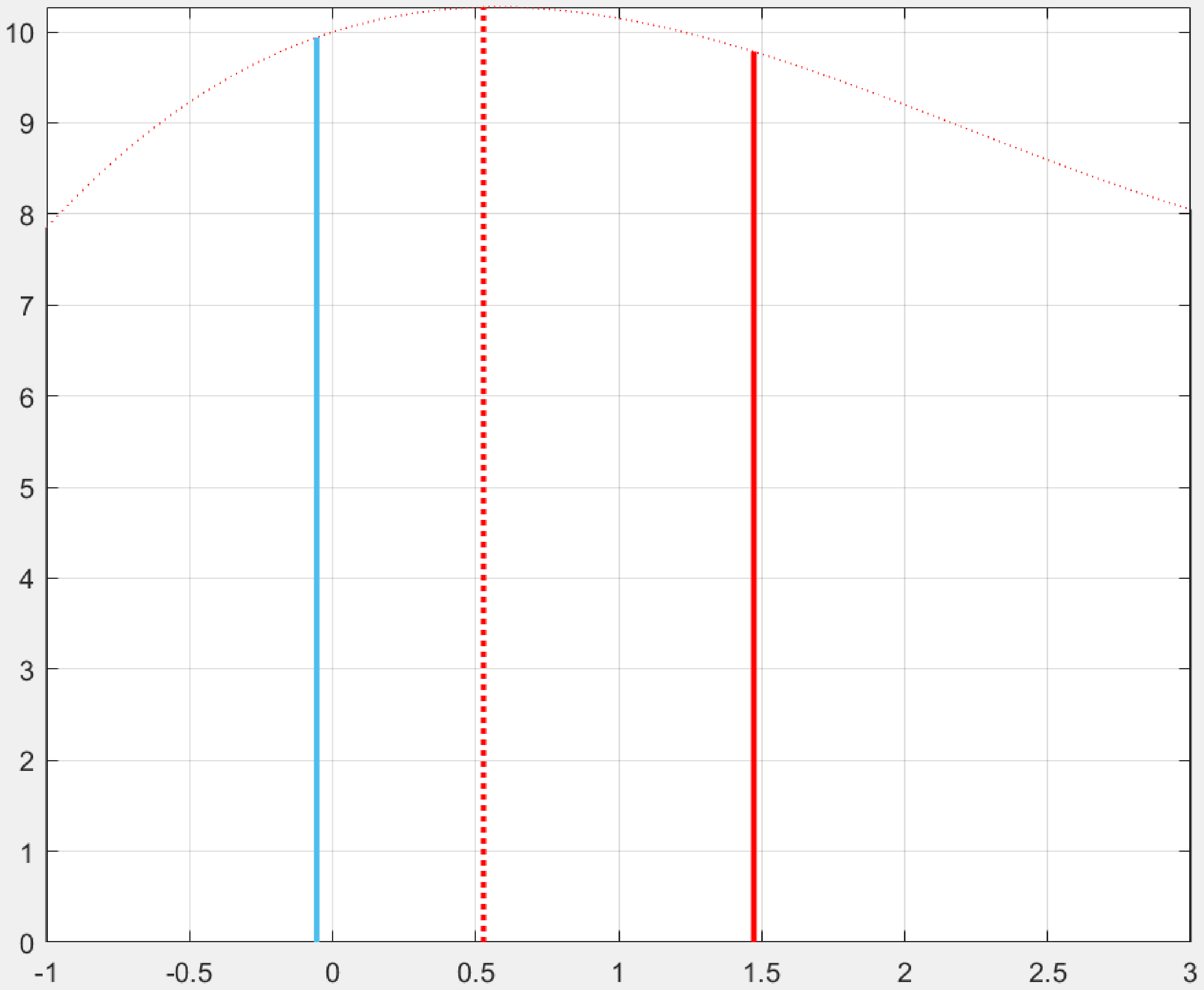
$$R + 1 = \frac{1}{R}$$
$$R^2 + R - 1 = 0$$

$$R_1 = -1.618$$
$$R_2 = +0.618$$

$$\frac{a}{b} = 0.618$$

$$a = 0.382(a + b)$$




10
9
8
7
6
5
4
3
2
1
0
-1
-0.5
0
0.5
1
1.5
2
2.5
3

Example

- For gutter with this trapezoidal section:
- Find θ for maximum area

$$A = 2 \times \sin(\theta) \times 2 \times \cos(\theta) + 4 \times \sin(\theta)$$

$$A = 4 \times \sin(\theta)\,(1 + \cos(\theta))$$

Example

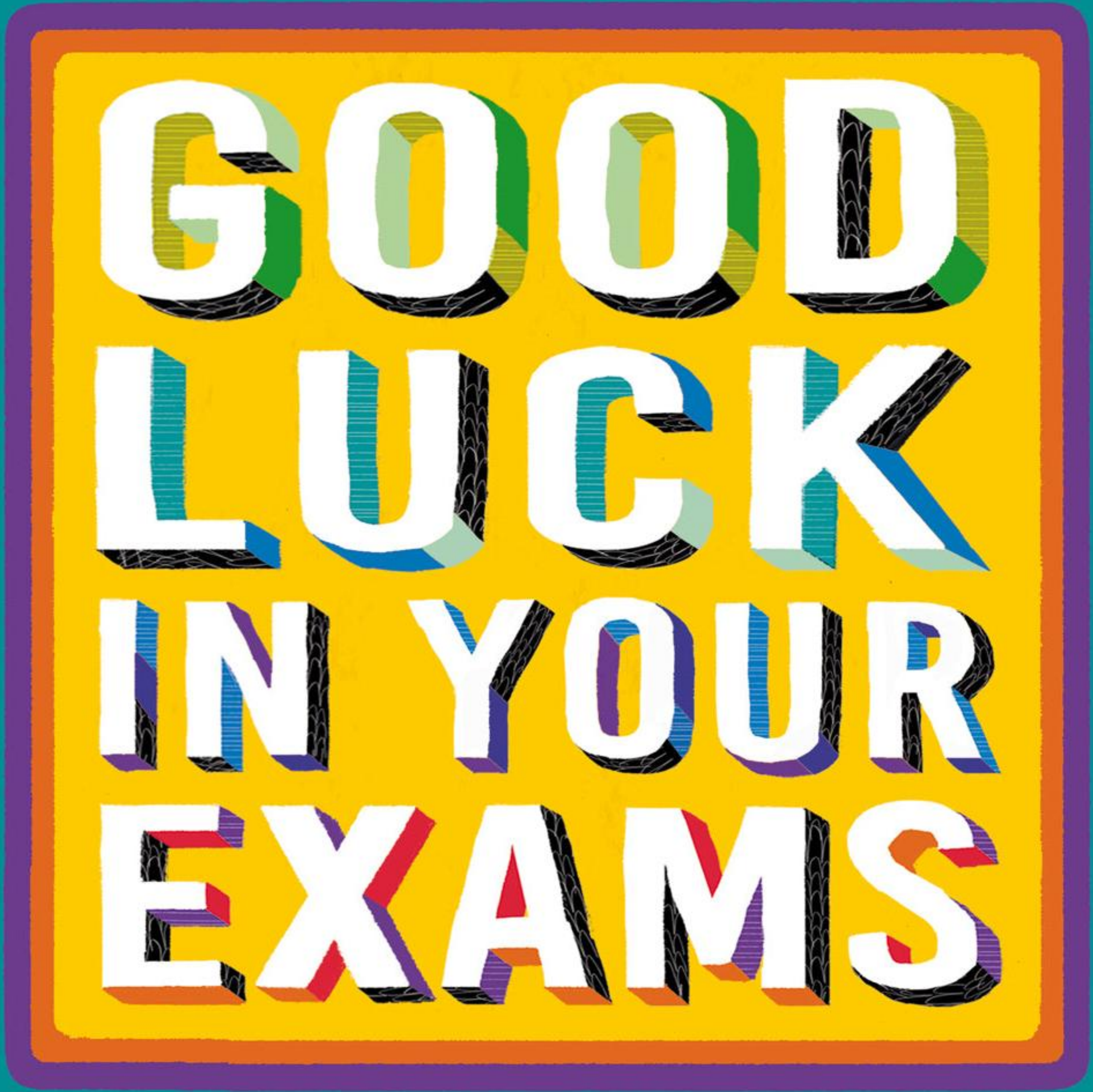
```
y = @(x) 4*sin(x)*(1+cos(x));
x1 = 0;     x2 = pi/2;
epsilon = 1e-4;
ratio = 0.382;
while abs(x2-x1) > epsilon
    y1 = y(x1);  y2 = y(x2);  d = x2-x1;
    x3 = x1 + ratio * d;
    x4 = x2 - ratio * d;
    y3 = y(x3);  y4 = y(x4);
    if (y3 > y4)
        x2 = x4;
    else
        x1 = x3;
    end
end
```

Example

```
y = @(x) 4*sin(x)*(1+cos(x));

x rad = 1.0472
x deg = 59.9999
y = 5.1962
```

Previous Exams


GOOD
LUCK
IN YOUR
EXAMS