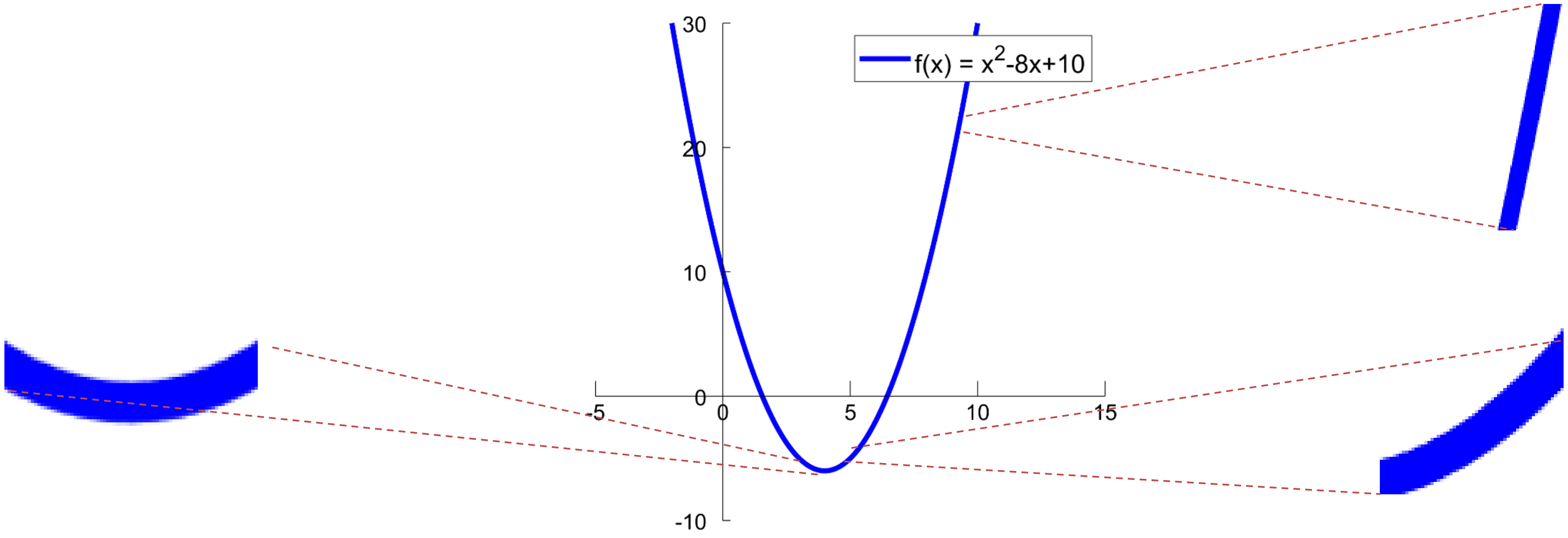
Gradient Descent Optimization

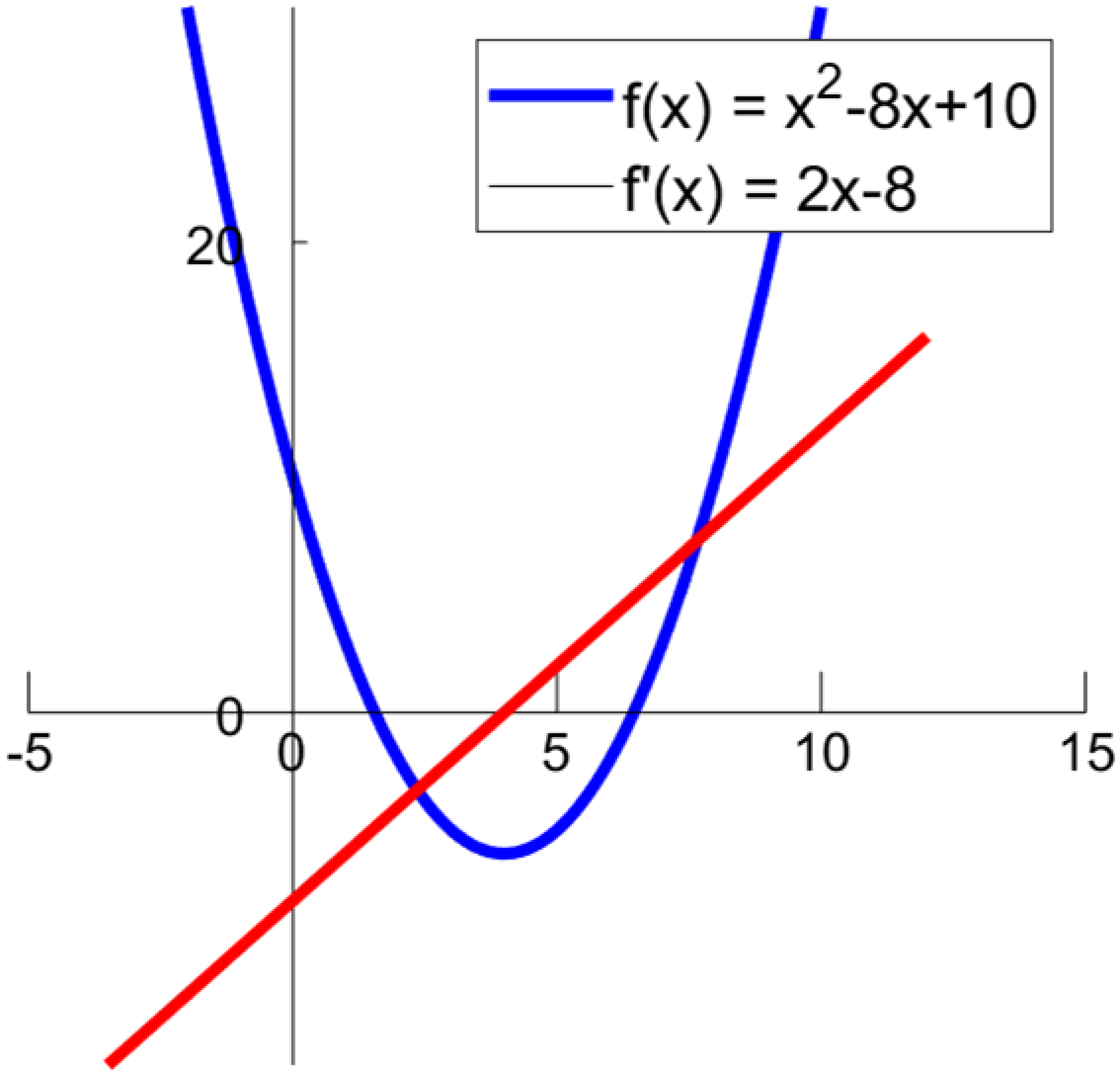
by Ahmet Sacan

Optimization

- Minimize a function f(x)
  - Maximization problems can trivially be restated as minimization.
- e.g., find $argmin(x^2 - 8x + 10)$



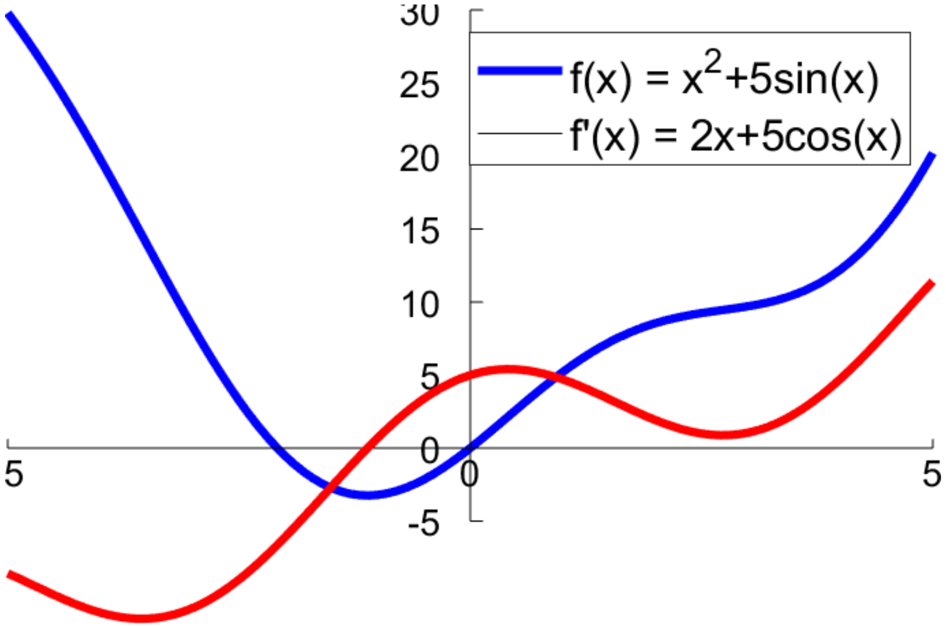
Optimization using Calculus

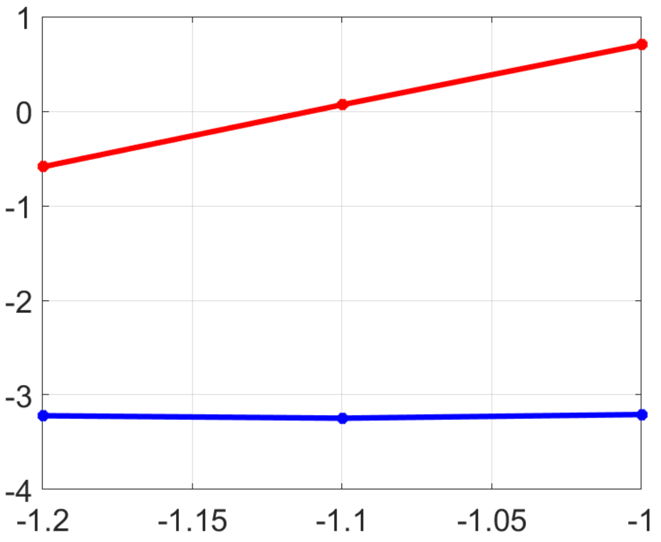


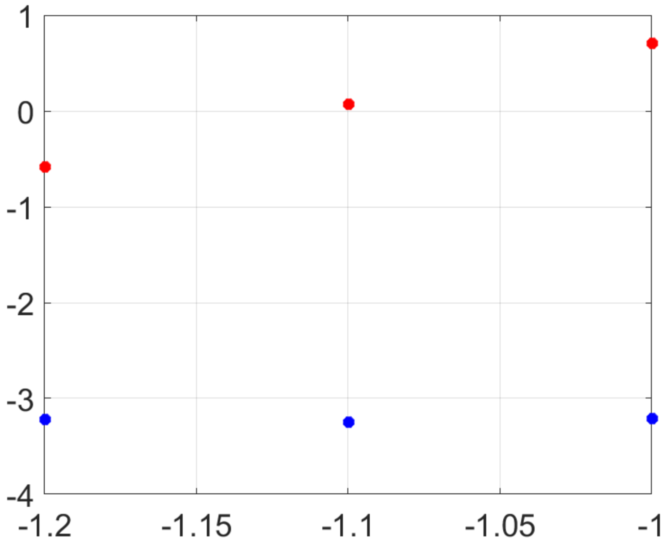
When (our) Calculus is not enough

$$f(x) = x^2 + 5\sin(x)$$

$$f'(x) = 2x + 5\cos(x)$$

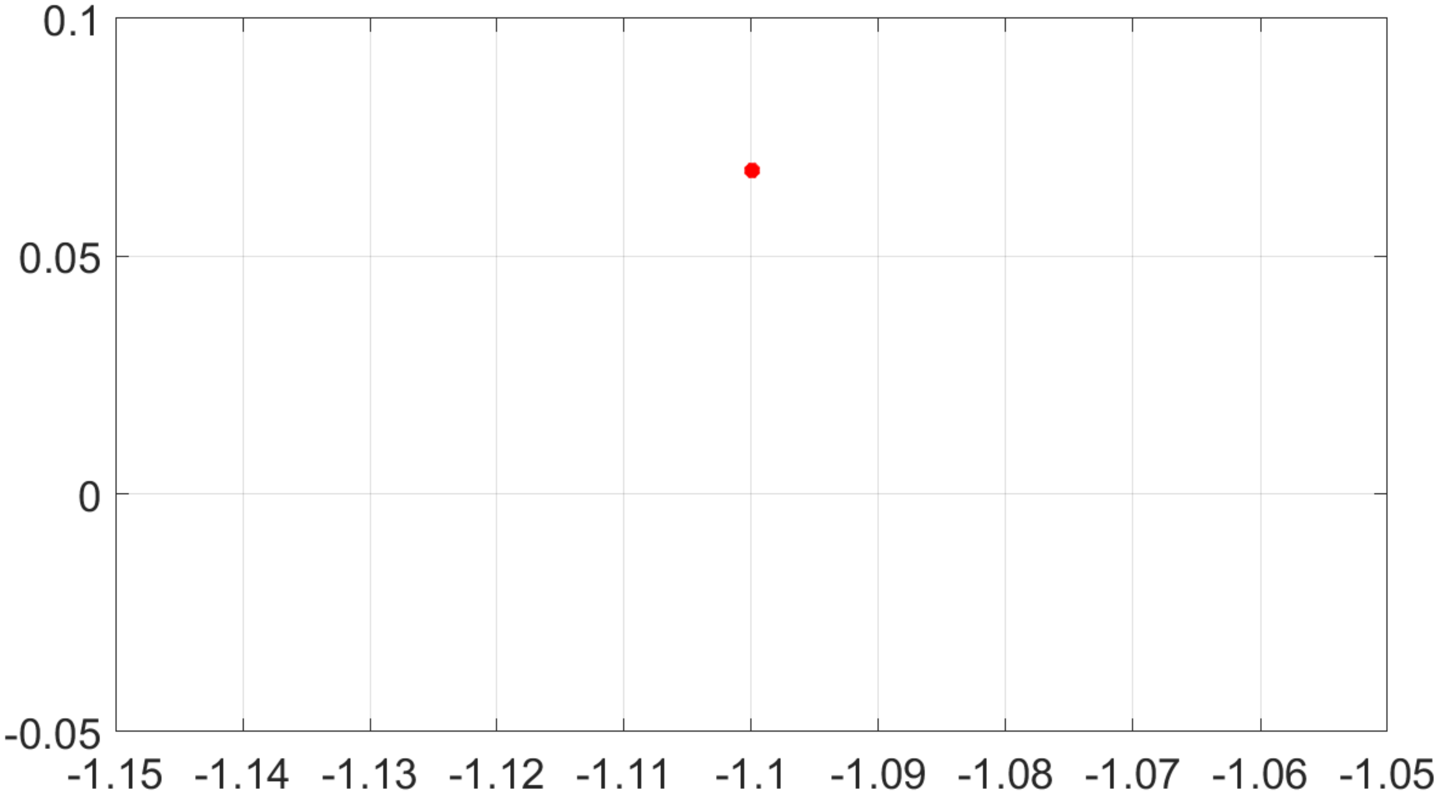






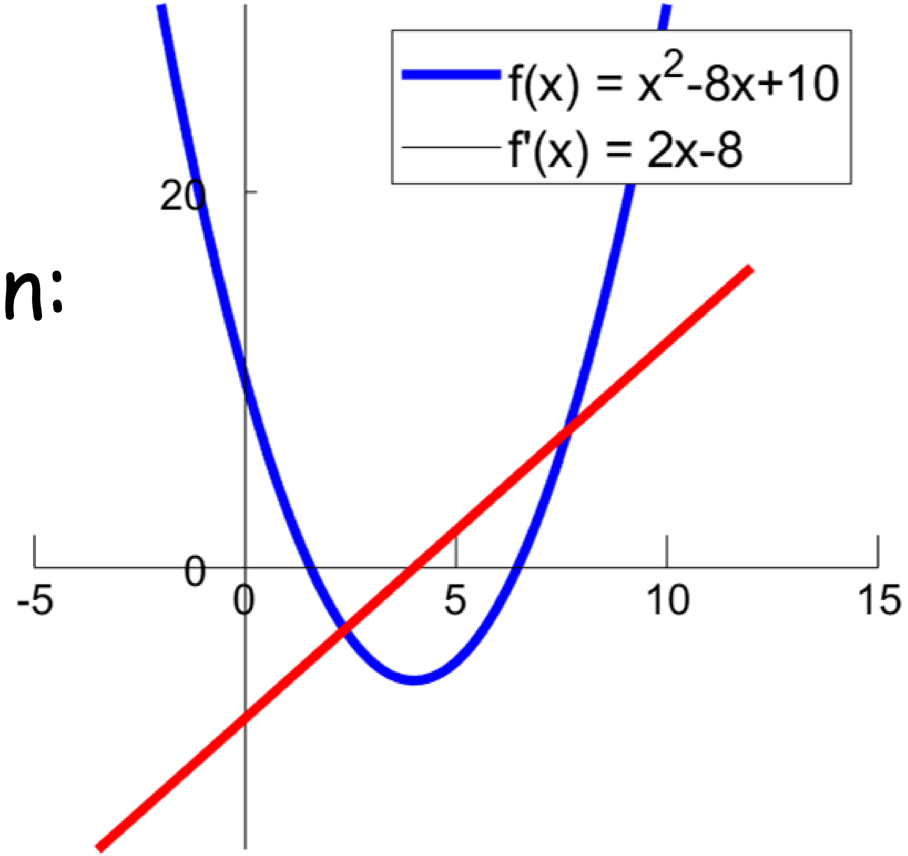
Zooming in

- $x = -1.110510504$
- $f'(x) = -0.0000000027142$



Gradient Descent

- A gradient (derivative) based optimization method
- aka **Steepest Descent**
- Start with $x = x_0$.
- Calculate derivative: $f'(x)$.
- Take a step in the opposite direction:
  - step: $\Delta x = -\eta f'(x)$
  - next value of x: $x_{i+1} = x_i - \eta f'(x_i)$
  - η is the **learning rate**.
- Repeat until you are happy

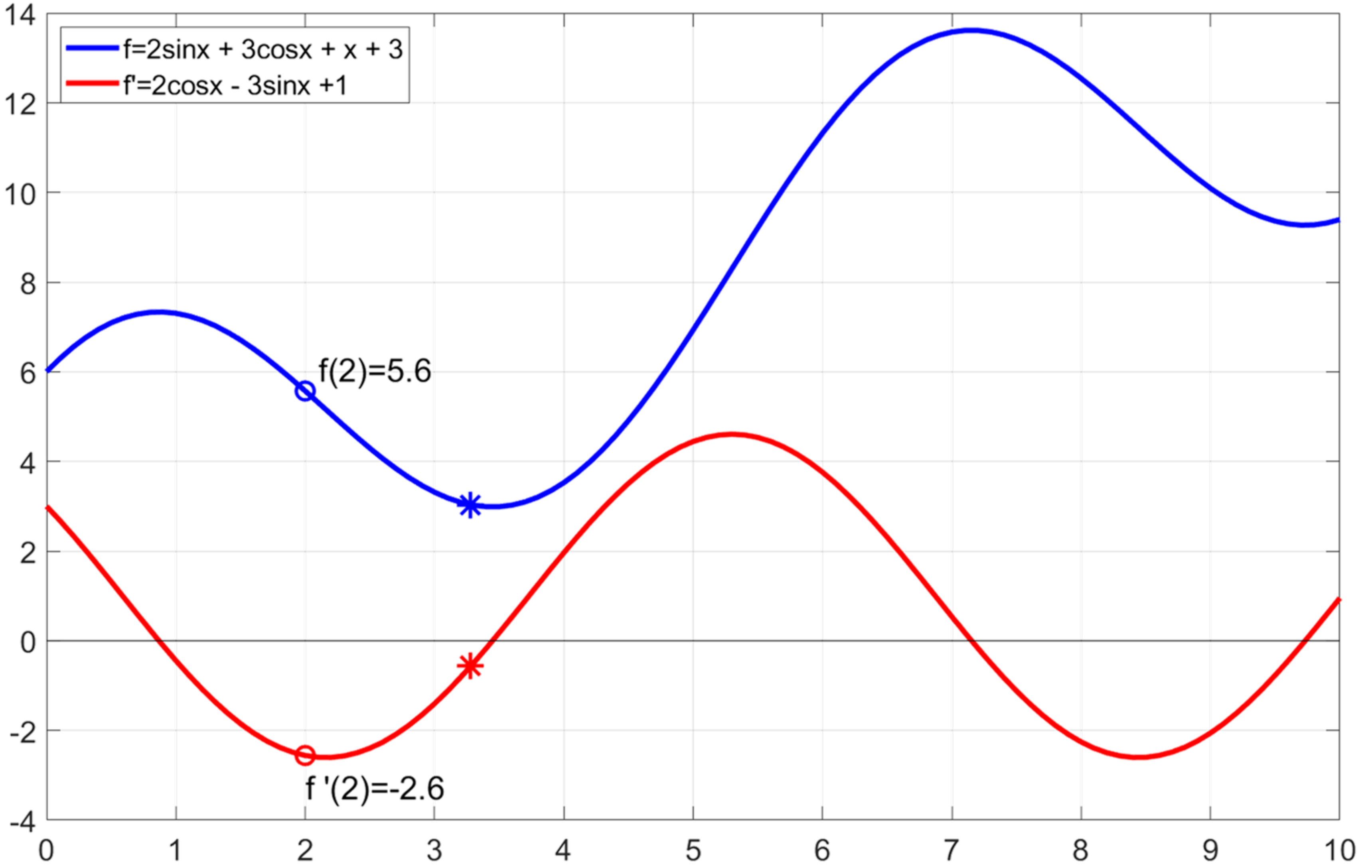


Gradient Descent Example

- $f(x) = 2sinx\ +\ 3cosx\ +\ x\ +\ 3$
- Start at initial guess $x_0 = 2$.
- Perform three iterations of gradient descent.
- Use $\eta = 0.5$

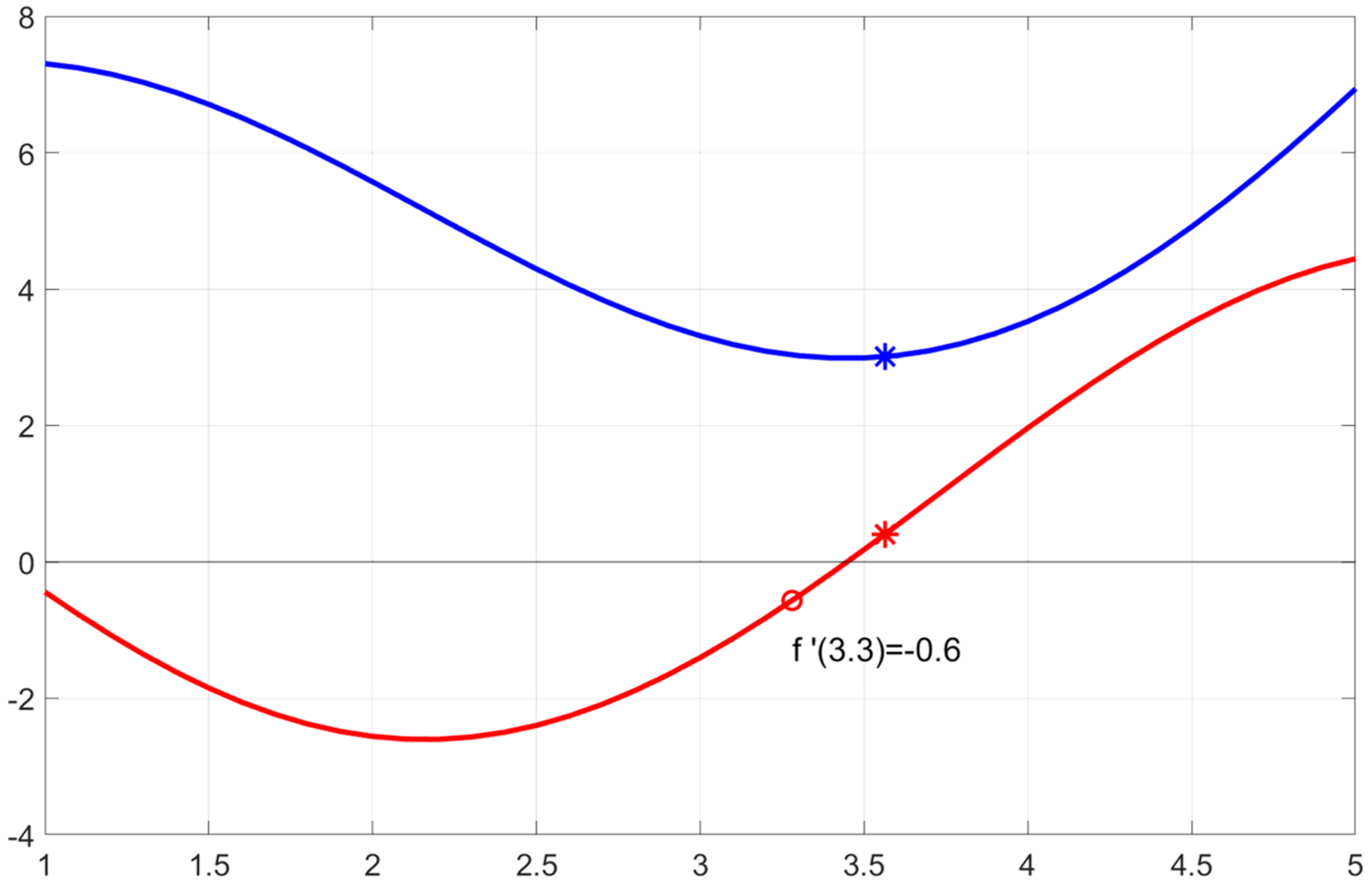
$$f(x) = 2sinx\ +\ 3cosx\ +\ x\ +\ 3$$
$$f'(x) = 2cosx\ - 3sinx\ + 1$$

- $f'(2) = -2.6$
- $\Delta x = -\eta f'(2)$
- $\quad = +1.3$
- $x_1 = x_0 + \Delta x$
- $\quad = 3.3$



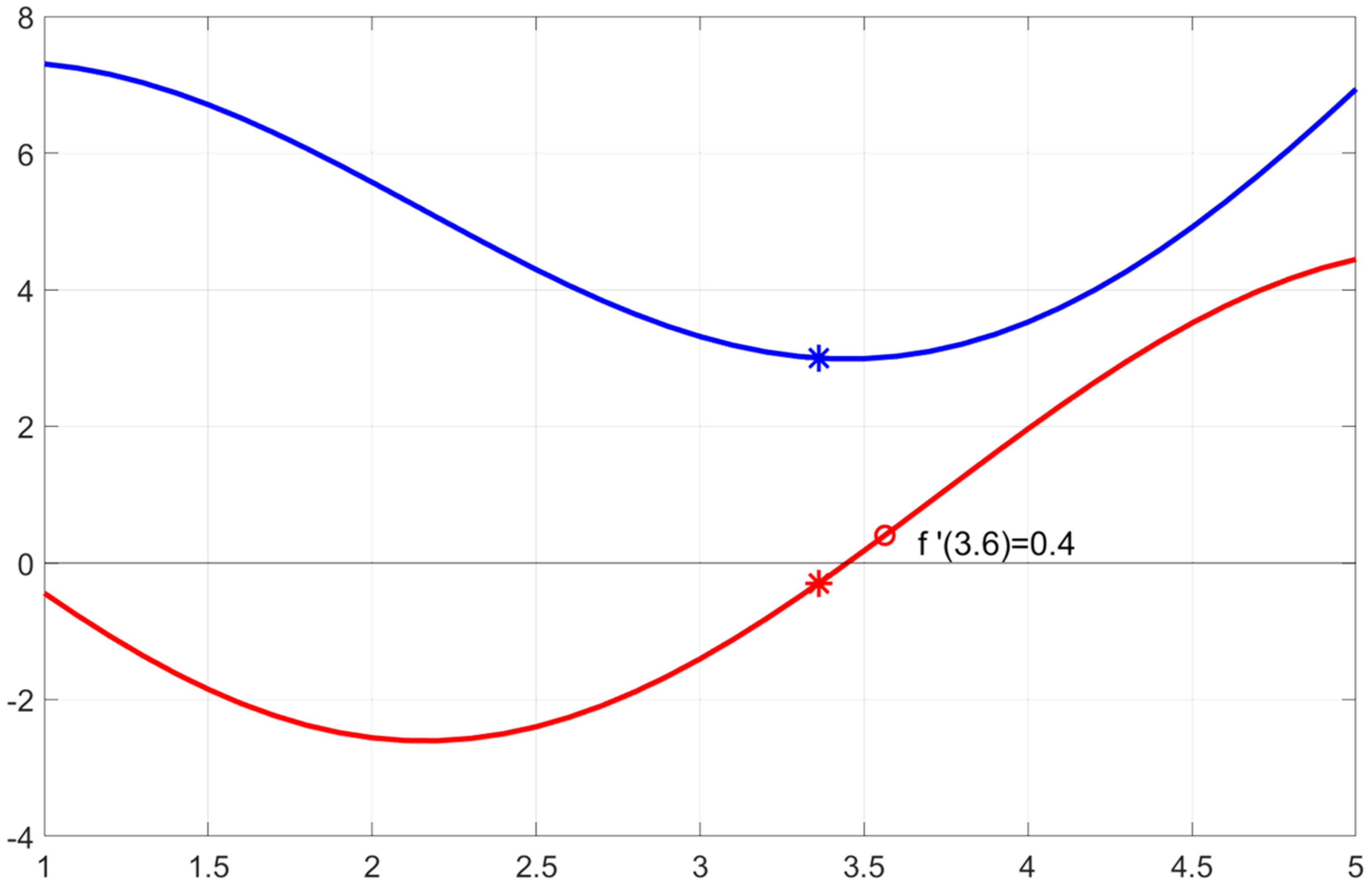
$$f(x) = 2sinx + 3cosx + x + 3$$
$$f'(x) = 2cosx - 3sinx + 1$$

- $f'(3.3) = -0.6$
- $\Delta x = -\eta f'(3.3)$
- $\quad = +0.3$
- $x_2 = x_1 + \Delta x$
- $\quad = 3.6$



$$f(x) = 2sinx + 3cosx + x + 3$$
$$f'(x) = 2cosx - 3sinx + 1$$

- $f'(3.6) = +0.4$
- $\Delta x = -\eta f'(3.6)$
- $\quad = -0.2$
- $x_3 = x_2 + \Delta x$
- $\quad = 3.4$



Gradient Descent iterations

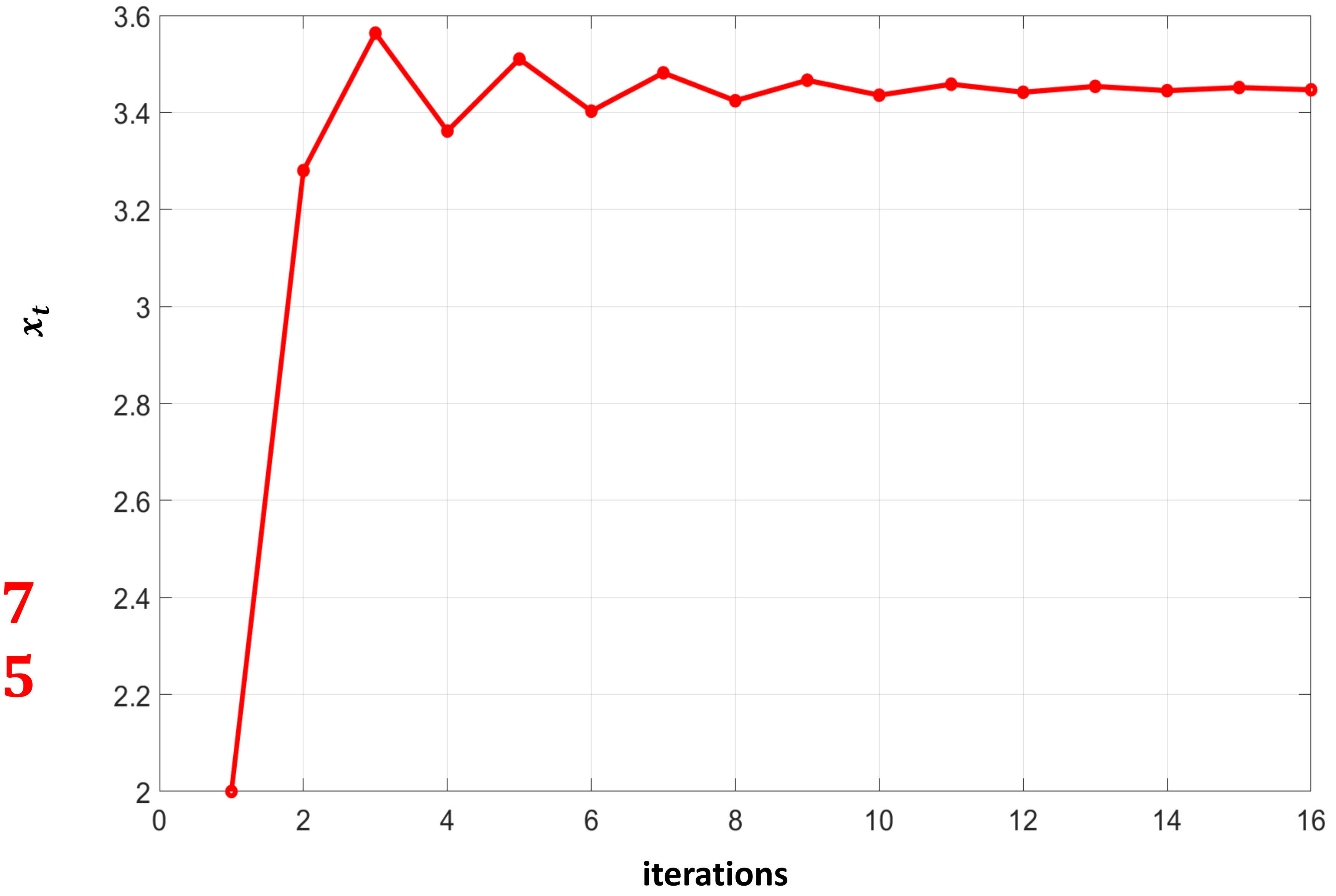
$x_0 = 2$
$x_1 = 3.28$
$x_2 = 3.56$

$\dots$

$x_{14} = 3.4\mathbf{51}$
$x_{15} = 3.4\mathbf{46}$

$\dots$

$x_{60} = 3.44856035\mathbf{7}$
$x_{61} = 3.44856035\mathbf{5}$

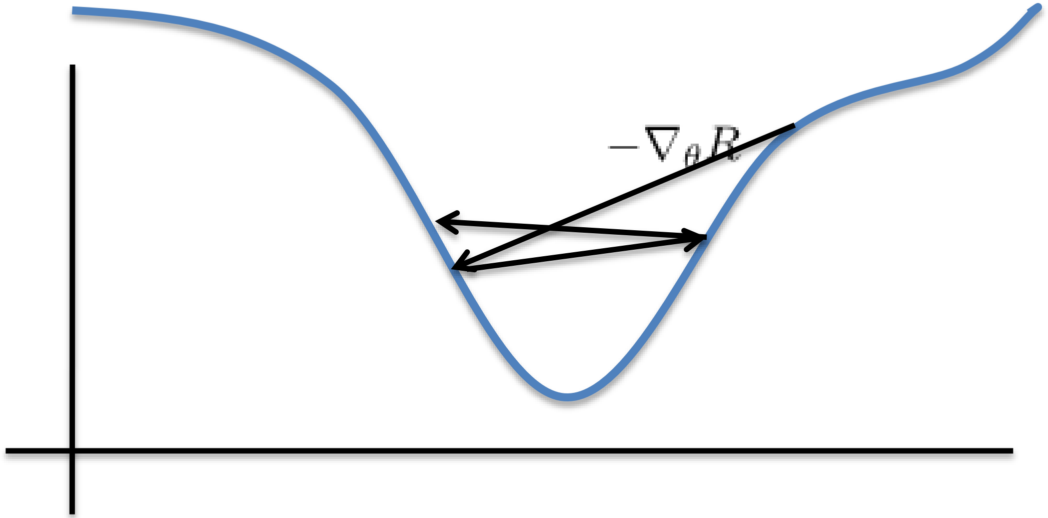


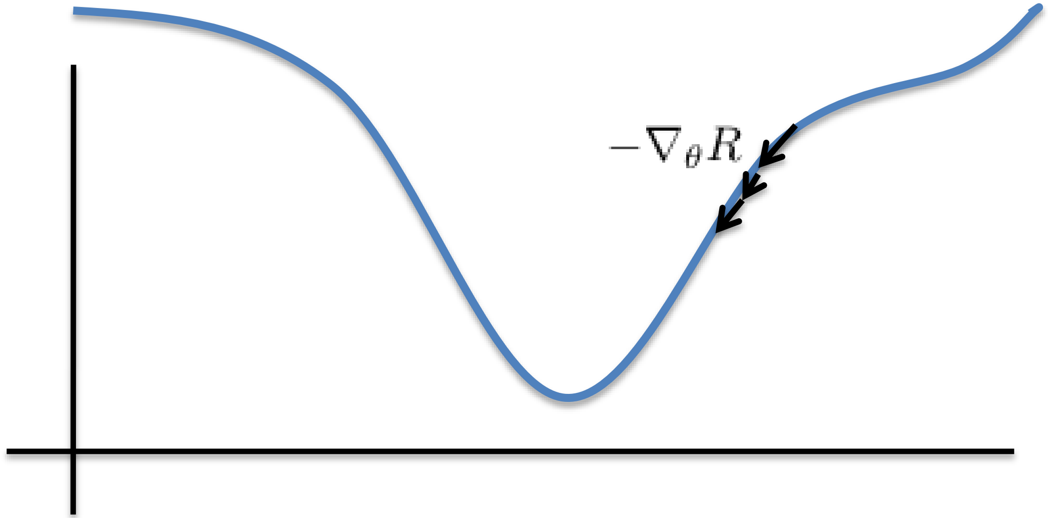
Stopping Criteria

- Stop when derivative becomes too small.
  - e.g., $f'(x_t) < 0.001$
- Stop when the change in function value is too small.
  - e.g., $\mathrm{f}(\mathrm{x}_{\mathrm{t+1}}) - \mathrm{f}(\mathrm{x}_{\mathrm{t}}) < 0.001$
- Stop when the step size is too small.
  - e.g., $\Delta x < 0.001$
- Stop after a fixed number of iterations.
  - e.g., 100 iterations.
- Any combination of the above.

Learning rate η

- Large η : risky
- Small η : slow





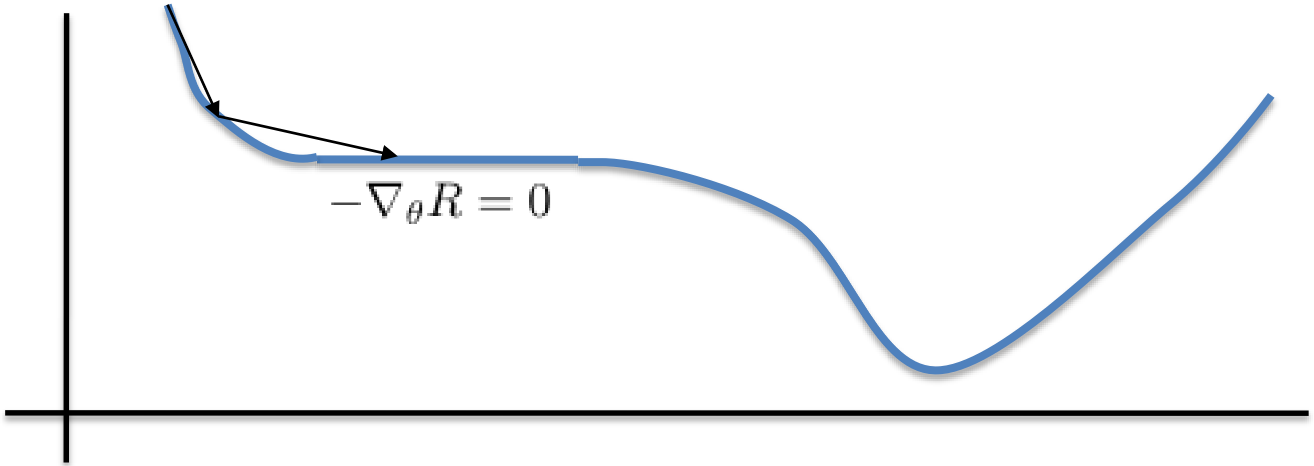
- Typical values: $\eta = 0.01$, $\eta = 0.001$

Adaptive learning rate

- Predefined schedule. e.g.,
  - Start with $\eta = 0.1$
  - Decrease η by 10% after each iteration.
- Reactive rules. e.g.,
  - If the derivative is in the same direction as the previous iteration, increase η by 10%.
  - If the derivative is not in the same as direction as the previous iteration, decrease η by 10%.

When gradient is zero

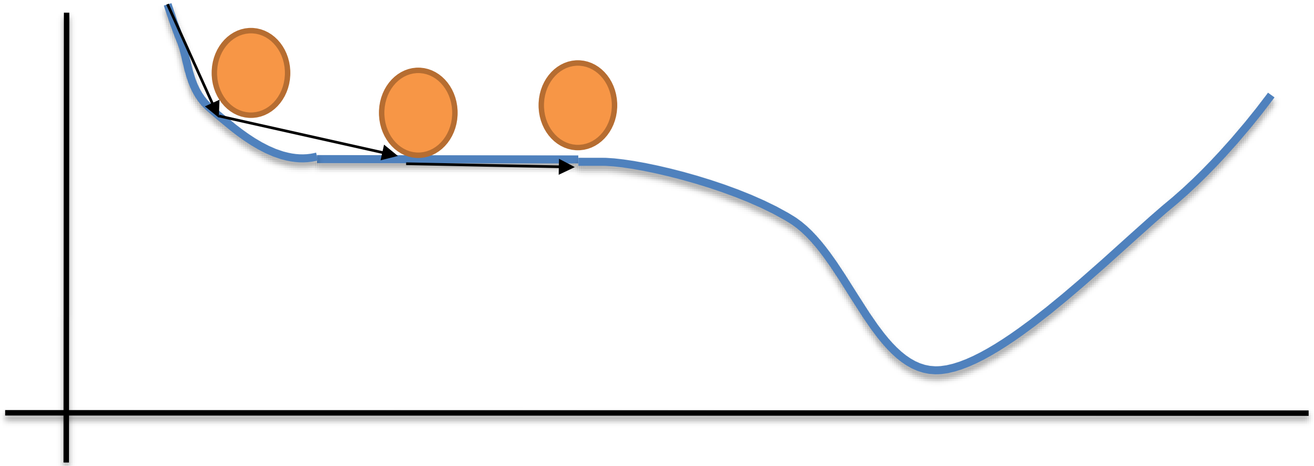
- Can stall if the gradient is ever 0 not at the minimum.



- Solution: momentum

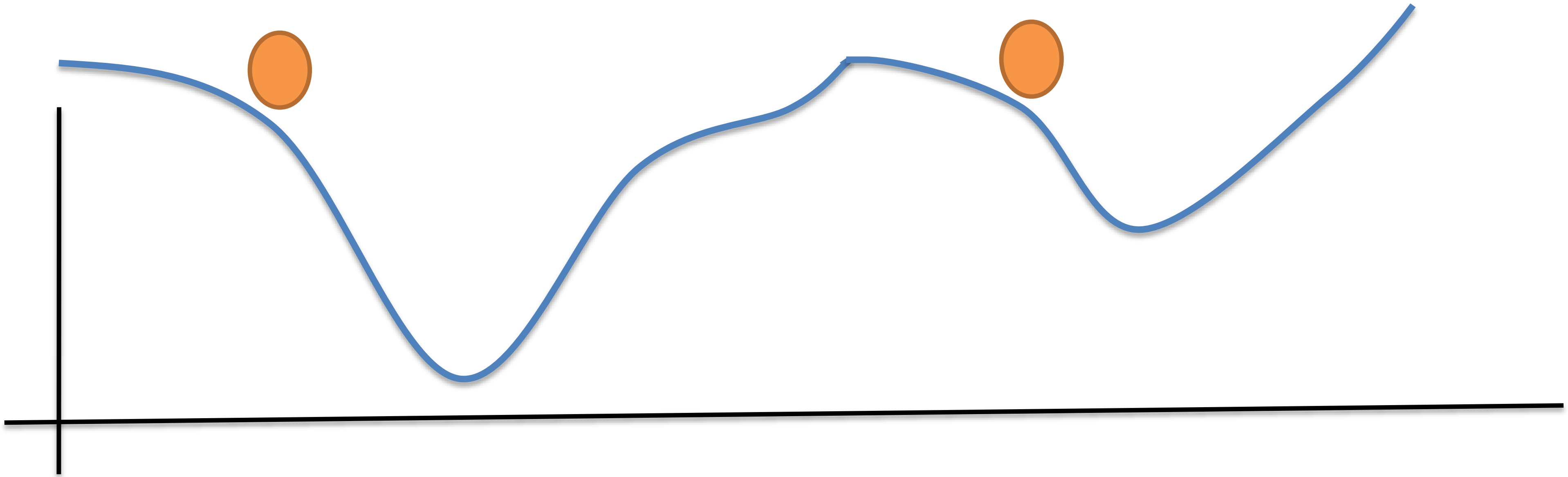
$$\Delta x = -\eta f'(x_t) + \gamma \Delta x_{t-1}$$

- Typical: $\gamma = 0.5 \sim 0.9$



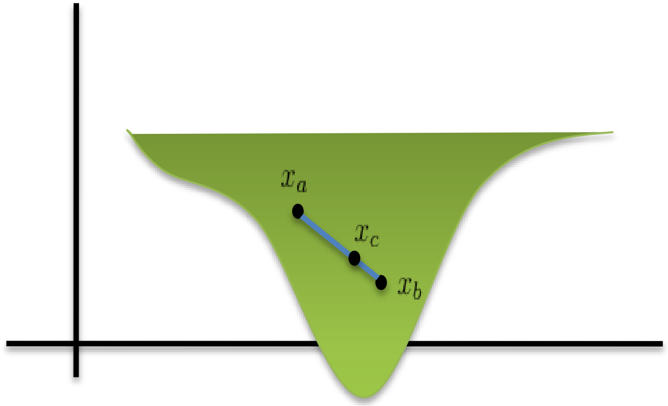
Local vs. Global Minima

- Local minima: The minima we can get to if we rolled down from a starting point
- Global minima: The lowest of all possible minimums.

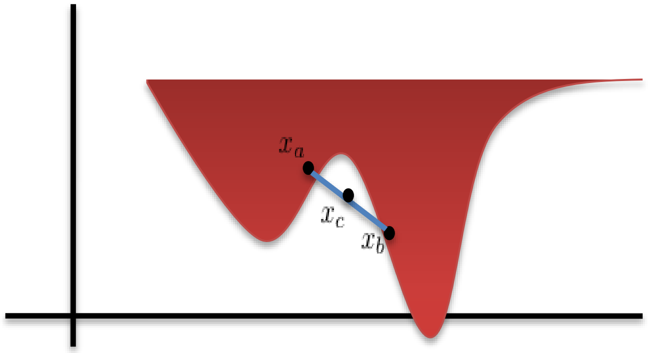


Convex Functions

- A function is convex if a line segment between any two points on the graph of the function lies above the graph.
- Any local minimum of a convex function is also a global minimum.



Convex



Non-Convex

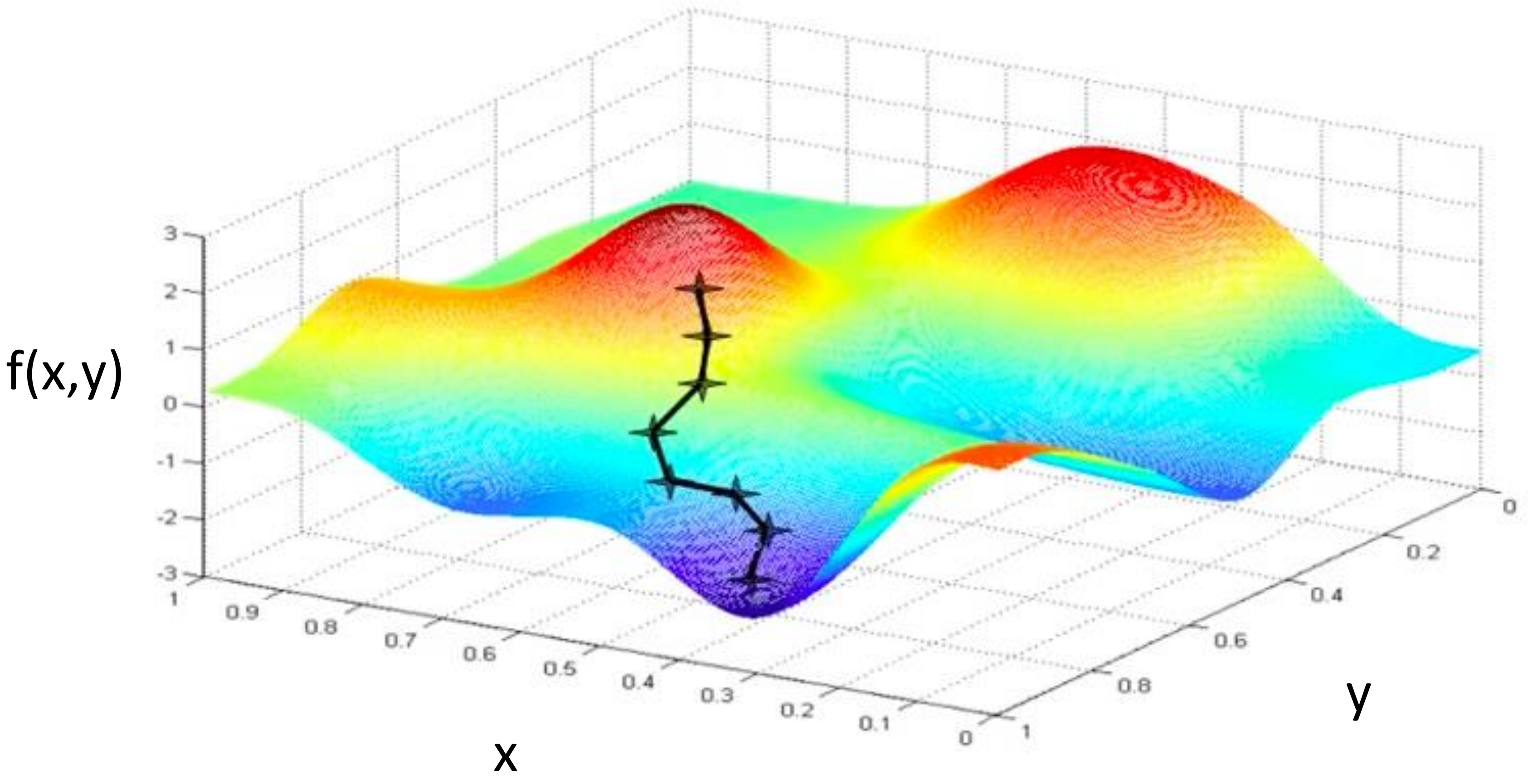
Gradient Descent in higher dimensions

- In higher dimensions, use **partial derivatives** (aka **gradient field**)

$$\nabla f = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \\ \vdots \end{bmatrix}$$

- $f(x, y) = x^2 + 3xy + 7x - y^2$
- $\nabla f = \begin{bmatrix} 2x + 3y + 7 \\ 3x - 2y \end{bmatrix}$
- Let $x_0 = 4$, $y_0 = 5$, $\eta = 0.5$
- $\nabla f = \begin{bmatrix} 2 * 4 + 3 * 5 + 7 \\ 3 * 4 - 2 * 5 \end{bmatrix} = \begin{bmatrix} 30 \\ 2 \end{bmatrix}$
- $\begin{bmatrix} x_1 \\ y_1 \end{bmatrix} = \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} - \eta \nabla f = \begin{bmatrix} 4 \\ 5 \end{bmatrix} - .5 \begin{bmatrix} 30 \\ 2 \end{bmatrix} = \begin{bmatrix} -11 \\ 4 \end{bmatrix}$

Gradient Descent in 2 dimensions



https://i.stack.imgur.com/w7ARo.png

When goal is to replicate g(x)

- Find the parameters θ such that $f_\theta(x)$ has the same/similar value as $g(x)$

- **Error/cost function:**

$$E_\theta(x) = \left(f_\theta(x) - g(x)\right)^2$$

- Redefined the problem:
  – Find the parameters θ such that $E_\theta(x)$ is minimized
- Minimize $E_\theta(x)$ using gradient descent.

When derivative is not available

- f(x) is available, but f'(x) is not.
- One solution:
  - Estimate f'(x) using: $\frac{f(x+\epsilon)-f(x)}{\epsilon}$

- Another solution:
  - Calculate $f(x+\epsilon)$ and $f(x-\epsilon)$
  - Move in the direction of the smaller one.

Related terms & methods

- Function: objective, cost, error, loss, energy
- Steepest ascent, hill-climbing
- Line search
- First order method

- Second order: Newton's method, conjugate gradient method

Conclusion

- Gradient descent
  - easy to implement
  - slow convergence
  - choice of learning rate
  - local minima
- Most programming libraries implement improved alternatives
  - Newton's method