Optimization (Introduction)

Optimization

Goal: Find the **minimizer** $\boldsymbol{x}^*$ that minimizes the **objective (cost) function** $f(\boldsymbol{x}): \mathcal{R}^n \to \mathcal{R}$

Unconstrained Optimization

Optimization

Goal: Find the **minimizer** $\boldsymbol{x}^*$ that minimizes the **objective (cost) function** $f(\boldsymbol{x}): \mathcal{R}^n \rightarrow \mathcal{R}$

Constrained Optimization

Unconstrained Optimization

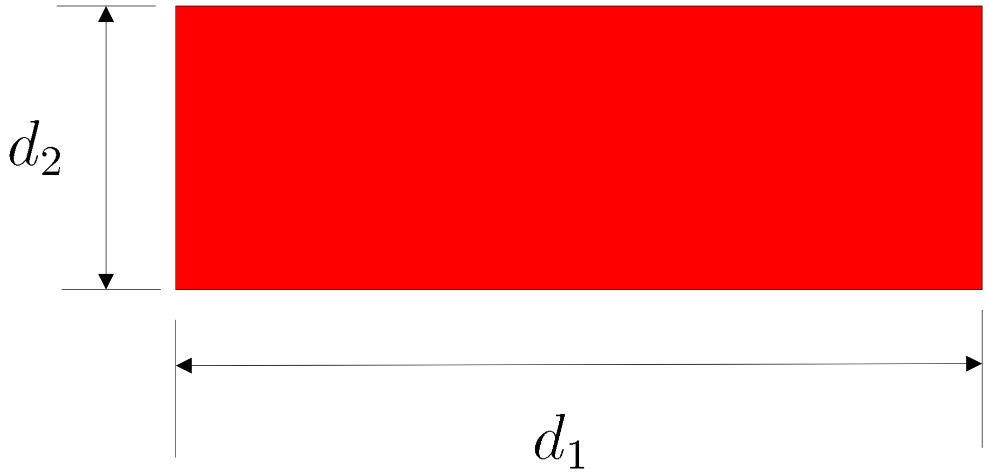
- **What if we are looking for a maximizer $\boldsymbol{x}^*$?**

$$f(\boldsymbol{x}^*) = \max_{\boldsymbol{x}} f(\boldsymbol{x})$$

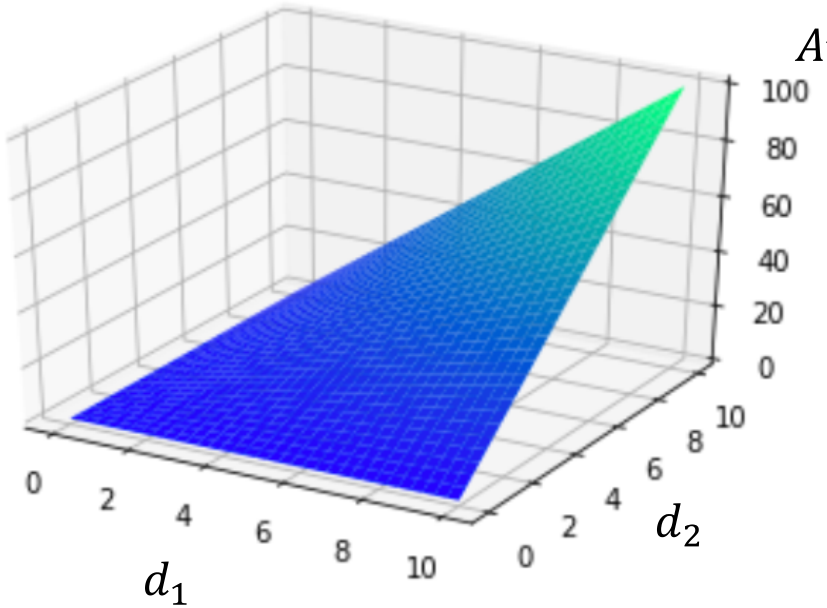
Calculus problem: maximize the rectangle area subject to perimeter constraint

$$\max_{\boldsymbol{d} \in \mathcal{R}^2} \quad f(d_1, d_2) = d_1 \times d_2$$

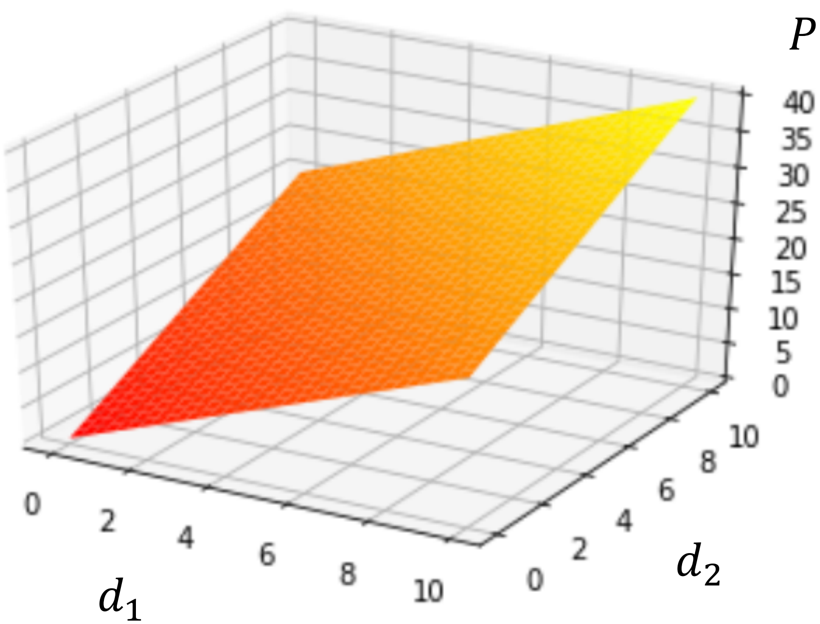
$$\text{such that} \quad g(d_1, d_2) = 2\,(d_1 + d_2) - 20 \leq 0$$

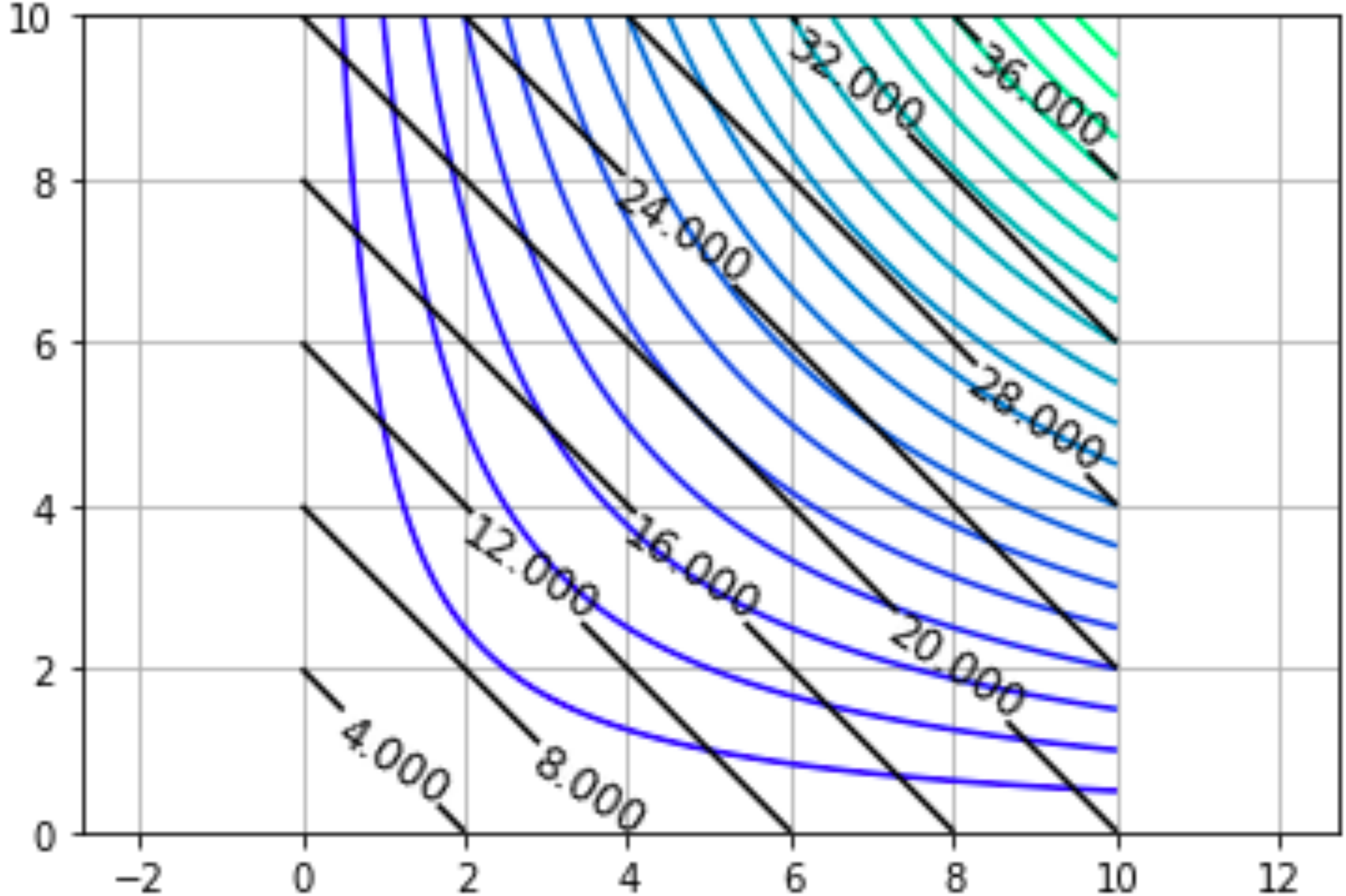


$Area = d_1 d_2$



$Perimeter = 2(d_1 + d_2)$





10
8
6
4
2
0
−2
0
2
4
6
8
10
12
4.000
8.000
12.000
16.000
20.000
24.000
28.000
32.000
36.000

What is the optimal solution? (1D)

$$f(x^*) = \min_x f(x)$$

(First-order) Necessary condition

(Second-order) Sufficient condition

Types of optimization problems

$$f(x^*) = \min_x f(x)$$

f: nonlinear, continuous and smooth

Gradient-free methods

Evaluate $f(x)$

Gradient (first-derivative) methods

Evaluate $f(x), f'(x)$

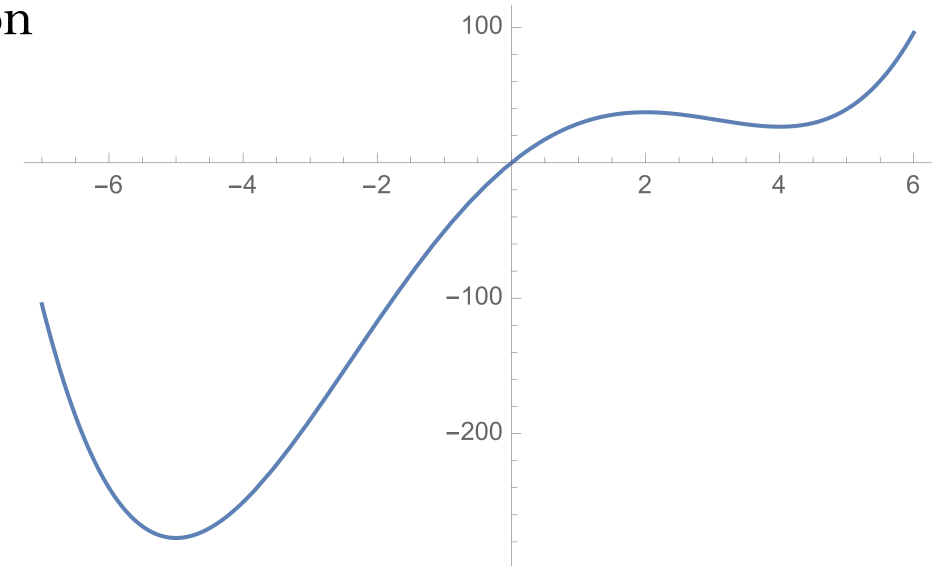
Second-derivative methods

Evaluate $f(x), f'(x), f''(x)$

Does the solution exists? Local or global solution?

Example (1D)

Consider the function $f(\boldsymbol{x}) = \frac{x^4}{4} - \frac{x^3}{3} - 11\,x^2 + 40x$. Find the stationary point and check the sufficient condition



What is the optimal solution? (ND)

$$f(\boldsymbol{x}^*) = \min_{\boldsymbol{x}} f(\boldsymbol{x})$$

(First-order) Necessary condition

1D: $f'(x) = 0$

(Second-order) Sufficient condition

1D: $f''(x) > 0$

Taking derivatives…

From linear algebra:

A symmetric $n \times n$ matrix $\boldsymbol{H}$ is **positive definite** if $\boldsymbol{y}^T \boldsymbol{H}\, \boldsymbol{y} > \boldsymbol{0}$ for any $\boldsymbol{y} \neq \boldsymbol{0}$

A symmetric $n \times n$ matrix $\boldsymbol{H}$ is **positive semi-definite** if $\boldsymbol{y}^T \boldsymbol{H}\, \boldsymbol{y} \geq \boldsymbol{0}$ for any $\boldsymbol{y} \neq \boldsymbol{0}$

A symmetric $n \times n$ matrix $\boldsymbol{H}$ is **negative definite** if $\boldsymbol{y}^T \boldsymbol{H}\, \boldsymbol{y} < \boldsymbol{0}$ for any $\boldsymbol{y} \neq \boldsymbol{0}$

A symmetric $n \times n$ matrix $\boldsymbol{H}$ is **negative semi-definite** if $\boldsymbol{y}^T \boldsymbol{H}\, \boldsymbol{y} \leq \boldsymbol{0}$ for any $\boldsymbol{y} \neq \boldsymbol{0}$

A symmetric $n \times n$ matrix $\boldsymbol{H}$ that is not negative semi-definite and not positive semi-definite is called **indefinite**

$$f(\boldsymbol{x}^*) = \min_{\boldsymbol{x}} f(\boldsymbol{x})$$

First order necessary condition: $\boldsymbol{\nabla} f(\boldsymbol{x}) = \boldsymbol{0}$

Second order sufficient condition: $\boldsymbol{H}(\boldsymbol{x})$ **is positive definite**

How can we find out if the Hessian is positive definite?

Types of optimization problems

$$f(\boldsymbol{x}^*) = \min_{\boldsymbol{x}} f(\boldsymbol{x})$$

f: nonlinear, continuous and smooth

Gradient-free methods

Evaluate $f(\boldsymbol{x})$

Gradient (first-derivative) methods

Evaluate $f(\boldsymbol{x}), \boldsymbol{\nabla} f(\boldsymbol{x})$

Second-derivative methods

Evaluate $f(\boldsymbol{x}), \boldsymbol{\nabla} f(\boldsymbol{x}), \boldsymbol{\nabla}^2 f(\boldsymbol{x})$

Example (ND)

Consider the function $f(x_1, x_2) = 2x_1^3 + 4x_2^2 + 2x_2 - 24x_1$

Find the stationary point and check the sufficient condition

Optimization (1D Methods)

Optimization in 1D: Golden Section Search

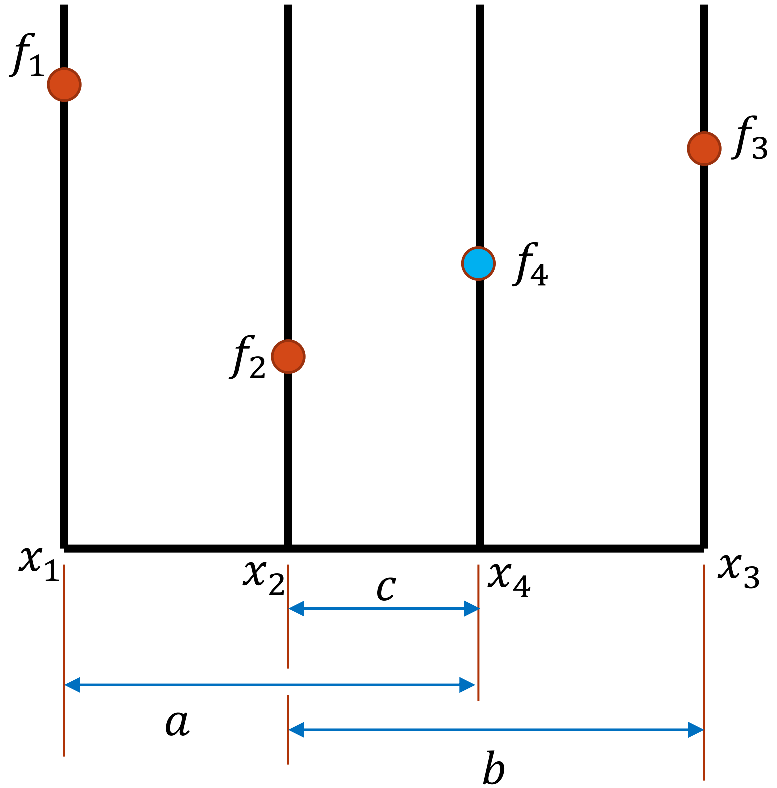
- Similar idea of bisection method for root finding
- Needs to bracket the minimum inside an interval
- Required the function to be unimodal

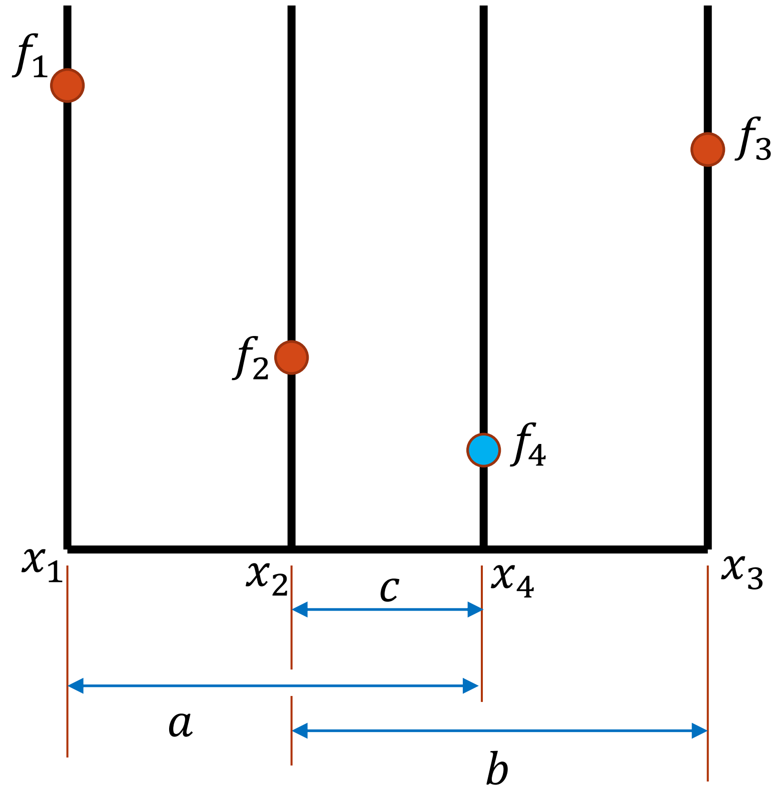
A function $f: \mathcal{R} \rightarrow \mathcal{R}$ is unimodal on an interval $[a, b]$

✓ There is a unique $\boldsymbol{x}^* \in [a, b]$ such that $f(\boldsymbol{x}^*)$ is the minimum in $[a, b]$

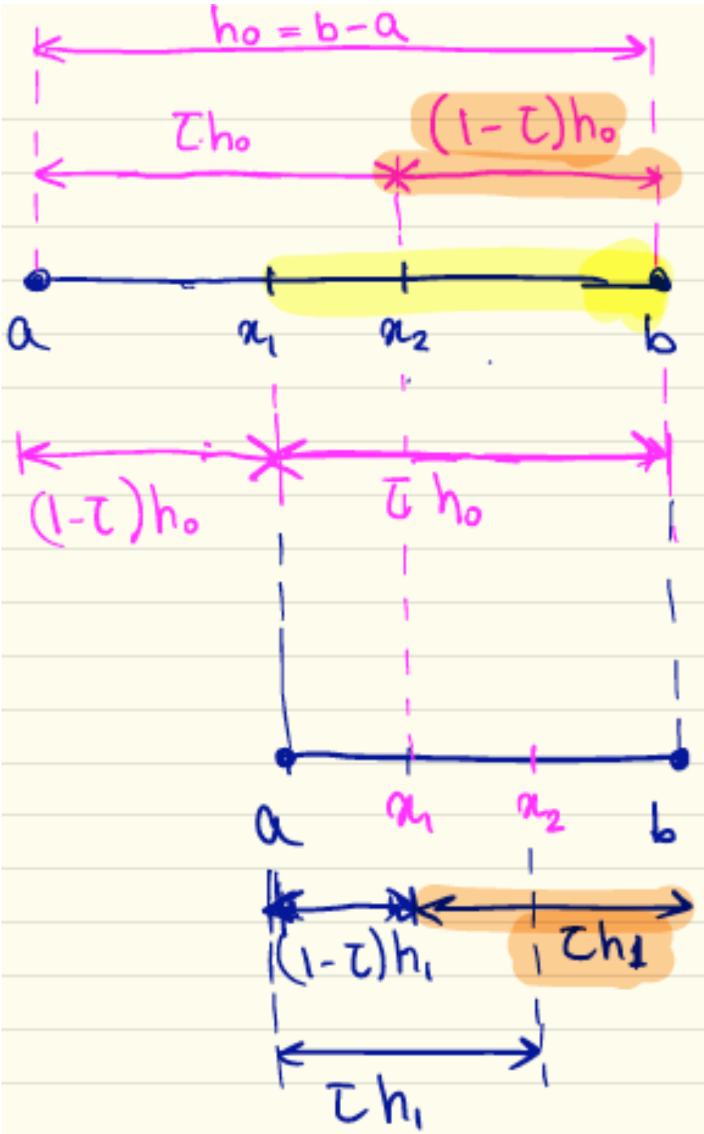
✓ For any $x_1, x_2 \in [a, b]$ with $x_1 < x_2$

- $x_2 < \boldsymbol{x}^* \Longrightarrow f(x_1) > f(x_2)$
- $x_1 > \boldsymbol{x}^* \Longrightarrow f(x_1) < f(x_2)$


f_1
f_2
f_3
f_4
x_1
x_2
x_4
x_3
c
a
b


f_1
f_2
f_3
f_4
x_1
x_2
x_4
x_3
c
a
b

Golden Section Search



Propose:

$x_1 = a + (1-\tau)h_0$

$x_2 = a + \tau h_0$

Evaluate $f_1 = f(x_1)$

$f_2 = f(x_2)$

if $(f_1 > f_2)$:

$a = x_1$

$x_1 = x_2$ → already have func. value!

$h_1 = b - a$

$x_2 = a + \tau h_1$

$f_2 = f(x_2)$ → only one

if $(f_1 < f_2)$:

$b = x_2$

$x_2 = x_1$

$x_1 = a + (1-\tau)h_1$

$f_1 = f(x_1)$

Golden Section Search

What happens with the length of the interval after one iteration?

$$h_1 = \tau\, h_o$$

Or in general: $h_{k+1} = \tau\, h_k$

Hence the interval gets reduced by $\boldsymbol{\tau}$

(for bisection method to solve nonlinear equations, τ=0.5)

For recursion:

$$\tau\, h_1 = (1 - \tau)\, h_o$$
$$\tau\, \tau\, h_o = (1 - \tau)\, h_o$$
$$\tau^2 = (1 - \tau)$$
$$\boldsymbol{\tau = 0.618}$$

Golden Section Search

- Derivative free method!

- Slow convergence:

$$\lim_{k\to\infty} \frac{|e_{k+1}|}{|e_k|} = 0.618 \quad r = 1 \; (linear\ convergence)$$

- Only one function evaluation per iteration

Example

Consider running golden section search on a function that is unimodal. If golden section search is started with an initial brakcet of $[-10, 10]$, what is the length of the new bracket after 1 iteration?

A) 20
B) 10
C) 12.36
D) 7.64

Newton's Method

Using Taylor Expansion, we can approximate the function f with a quadratic function about x_0

$$f(x) \approx f(x_0) + f'(x_0)\,(x - x_0) + \frac{1}{2} f''(x_0)\,(x - x_0)^2$$

And we want to find the minimum of the quadratic function using the first-order necessary condition

Newton's Method

- **Algorithm:**

$x_0 = \text{starting guess}$

$x_{k+1} = x_k - f'(x_k)/f''(x_k)$

- **Convergence:**
  - Typical quadratic convergence
  - Local convergence (start guess close to solution)
  - May fail to converge, or converge to a maximum or point of inflection

Newton’s Method (Graphical Representation)

Example

Consider the function $f(x) = 4\,x^3 + 2\,x^2 + 5\,x + 40$

If we use the initial guess $x_0 = 2$, what would be the value of x after one iteration of the Newton's method?

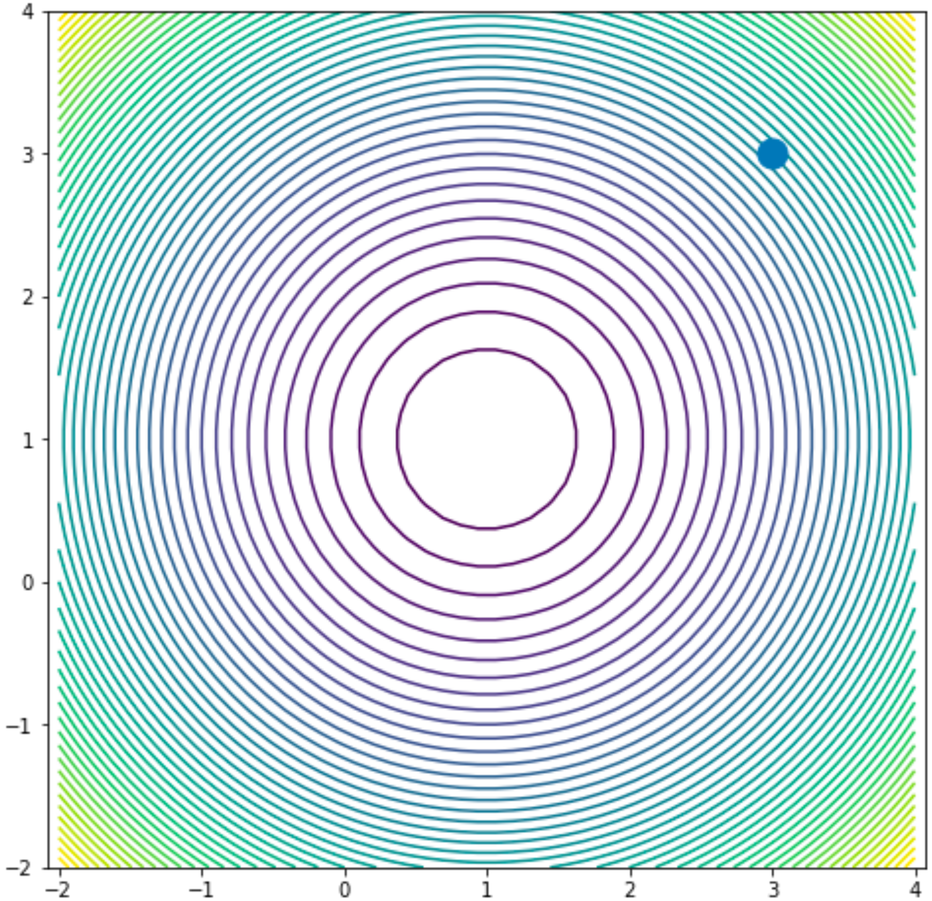
Optimization (ND Methods)

Optimization in ND: Steepest Descent Method

Given a function $f(\boldsymbol{x}): \mathcal{R}^n \rightarrow \mathcal{R}$ at a point $\boldsymbol{x}$, the function will decrease its value in the direction of steepest descent: $-\boldsymbol{\nabla} f(\boldsymbol{x})$

What is the steepest descent direction?

$$f(x_1, x_2) = (x_1 - 1)^2 + (x_2 - 1)^2$$



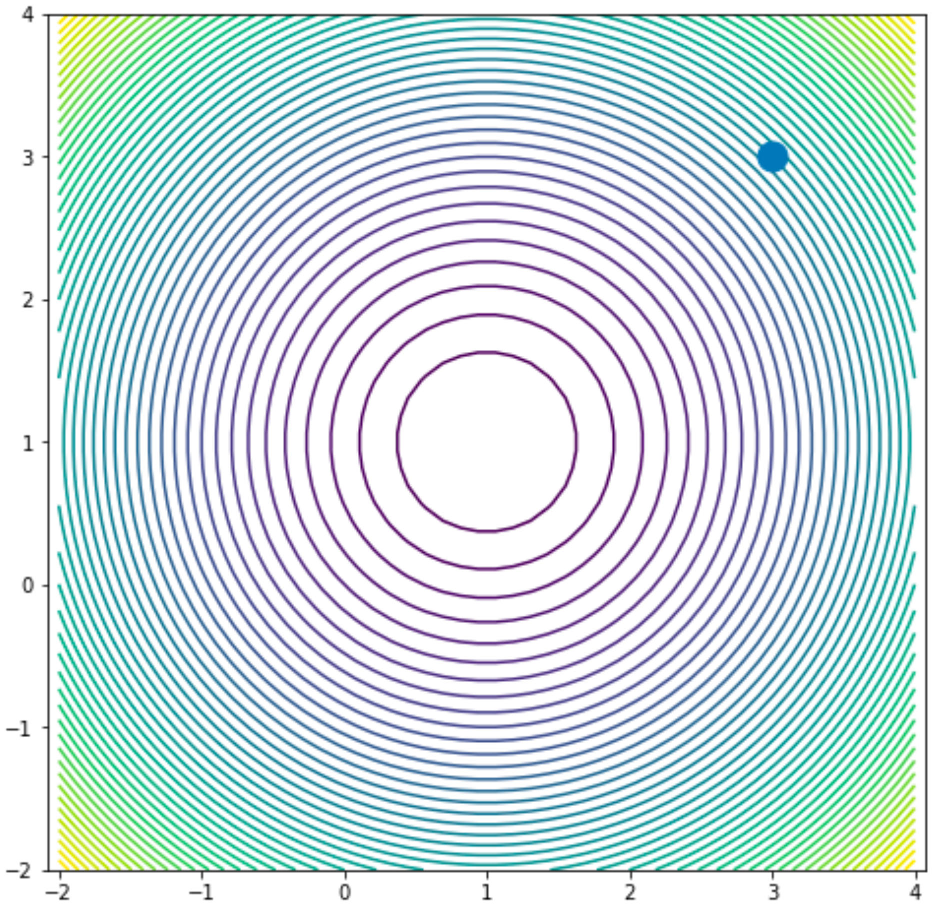
Steepest Descent Method

Start with initial guess:

$$\boldsymbol{x}_0 = \begin{bmatrix} 3 \\ 3 \end{bmatrix}$$

Check the update:

$$f(x_1, x_2) = (x_1 - 1)^2 + (x_2 - 1)^2$$



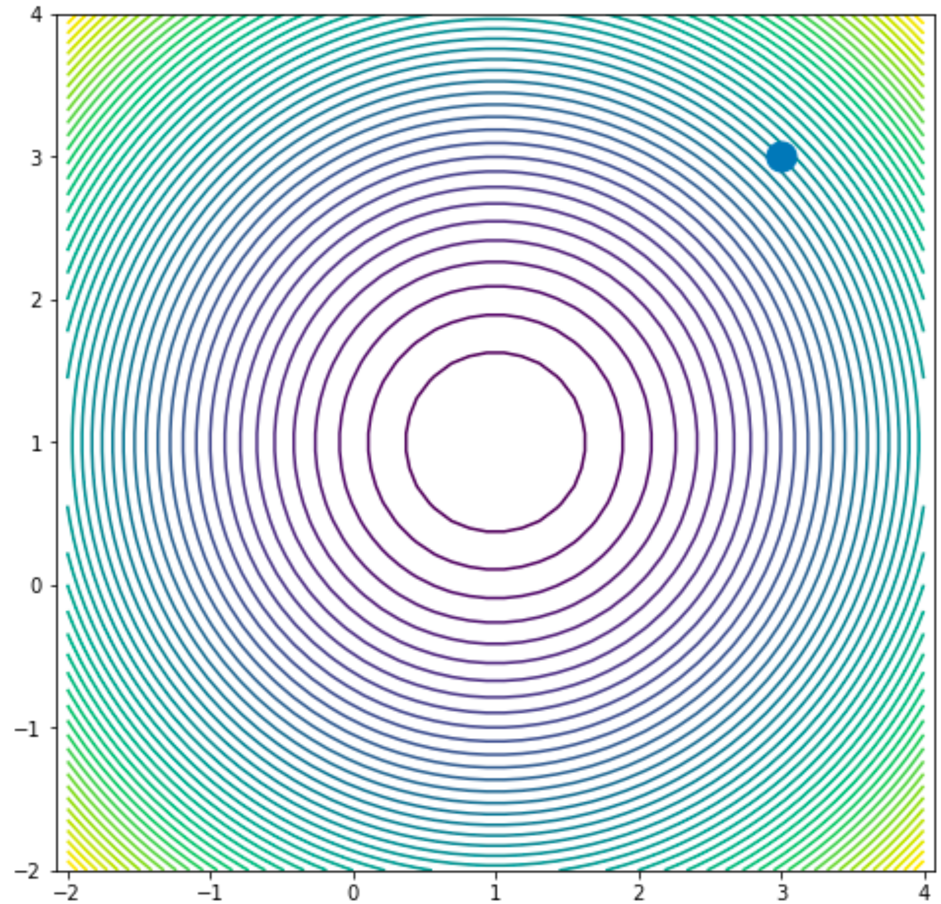
Steepest Descent Method

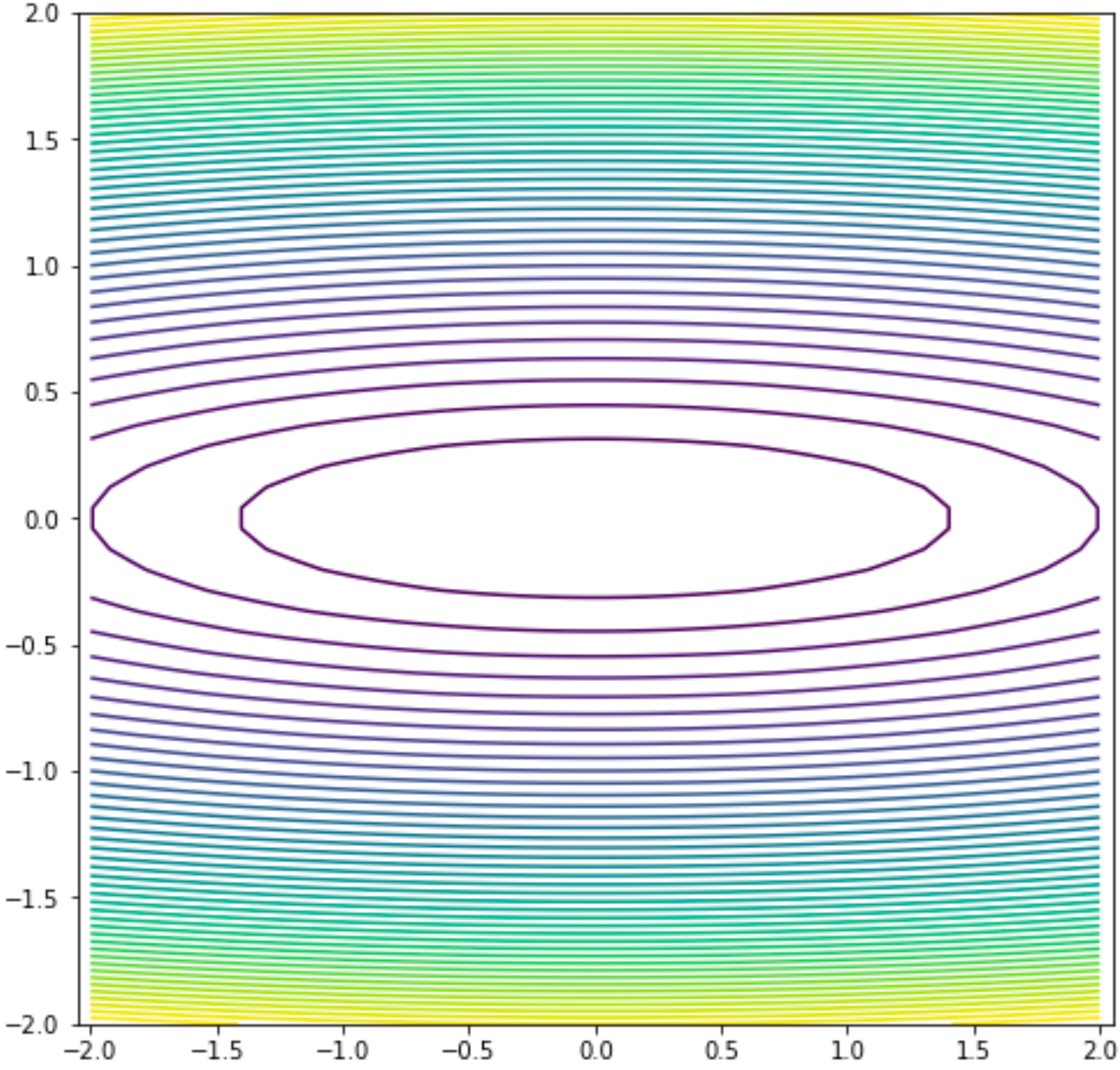
Update the variable with:

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \alpha_k \boldsymbol{\nabla} f(\boldsymbol{x}_k)$$

How far along the gradient should we go? What is the “best size” for α_k?

$$f(x_1, x_2) = (x_1 - 1)^2 + (x_2 - 1)^2$$





2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
−2.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0

Steepest Descent Method

Algorithm:

Initial guess: $\boldsymbol{x}_0$

Evaluate: $\boldsymbol{s}_k = -\boldsymbol{\nabla} f(\boldsymbol{x}_k)$

Perform a line search to obtain α_k (for example, Golden Section Search)

$$\alpha_k = \underset{\alpha}{\operatorname{argmin}} f(\boldsymbol{x}_k + \alpha\, \boldsymbol{s}_k)$$

Update: $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha_k\, \boldsymbol{s}_k$

Line Search

Example

Consider minimizing the function

$$f(x_1, x_2) = 10(x_1)^3 - (x_2)^2 + x_1 - 1$$

Given the initial guess

$$x_1 = 2, x_2 = 2$$

what is the direction of the first step of gradient descent?

Newton's Method

Using Taylor Expansion, we build the approximation:

Newton's Method

Algorithm:

Initial guess: $\boldsymbol{x}_0$

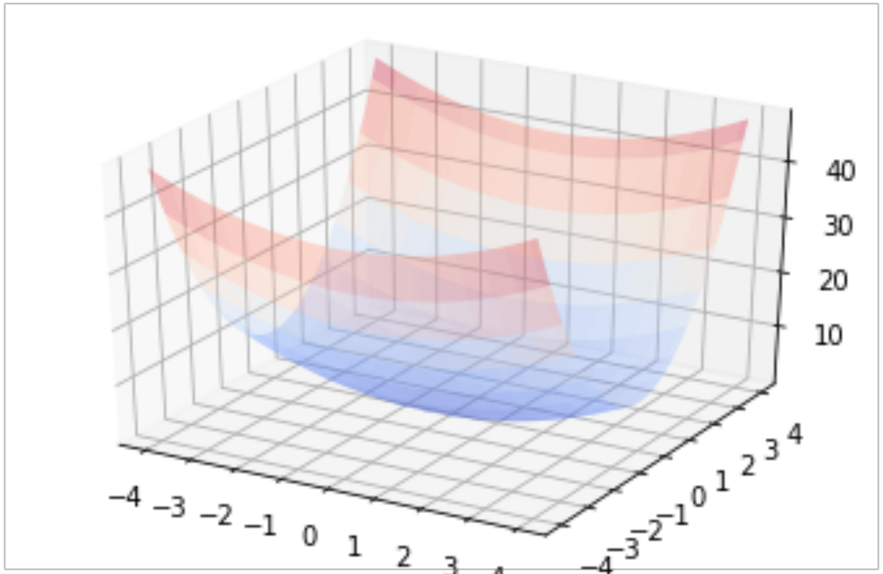
Solve: $\boldsymbol{H}_{\boldsymbol{f}}(\boldsymbol{x}_k)\,\boldsymbol{s}_k = -\boldsymbol{\nabla} f(\boldsymbol{x}_k)$

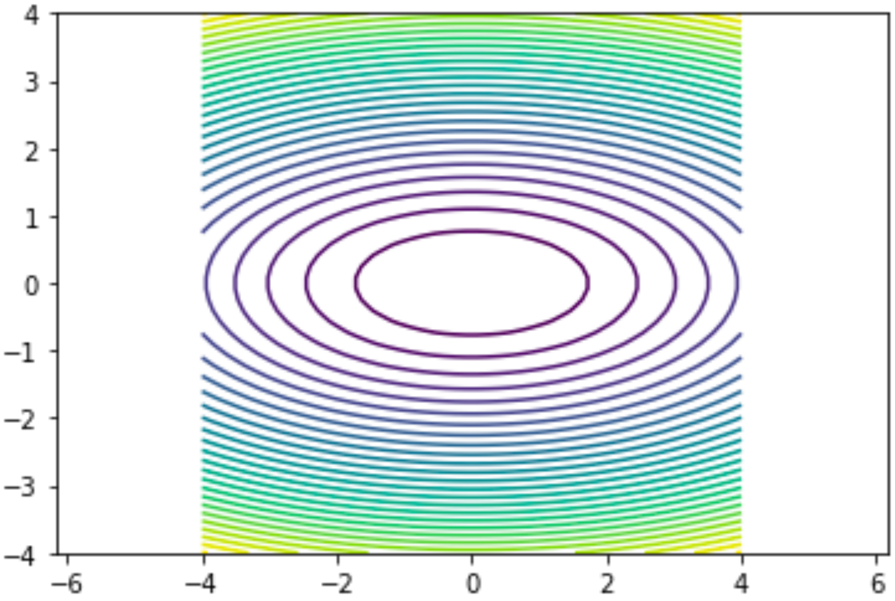
Update: $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \boldsymbol{s}_k$

Note that the Hessian is related to the curvature and therefore contains the information about how large the step should be.

Try this out!

$f(x, y) = 0.5x^2 + 2.5y^2$





When using the Newton's Method to find the minimizer of this function, estimate the number of iterations it would take for convergence?

A) 1 B) 2-5 C) 5-10 D) More than 10 E) Depends on the initial guess

Newton's Method Summary

Algorithm:

Initial guess: $\boldsymbol{x}_0$

Solve: $\boldsymbol{H_f}(\boldsymbol{x}_k)\,\boldsymbol{s}_k = -\nabla f(\boldsymbol{x}_k)$

Update: $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \boldsymbol{s}_k$

About the method…

- Typical quadratic convergence ☺
- Need second derivatives ☹
- Local convergence (start guess close to solution)
- Works poorly when Hessian is nearly indefinite
- Cost per iteration: $O(n^3)$