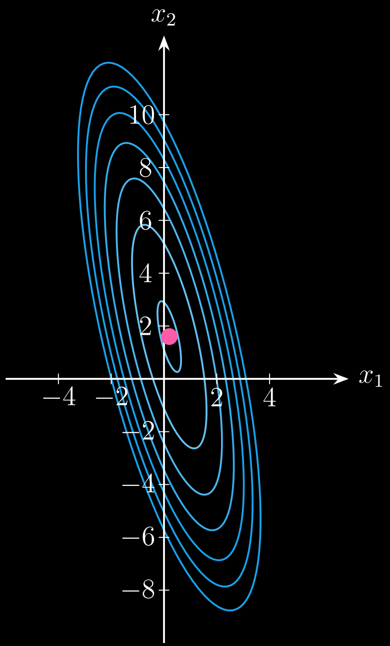
Quadratic optimization:

$$\min_{\boldsymbol{x}} \quad \frac{1}{2}\boldsymbol{x}^\top \boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}^\top \boldsymbol{x}$$

with

$$\boldsymbol{A} = \begin{bmatrix} 9 & 2 \\ 2 & 1 \end{bmatrix} \quad \text{and} \quad \boldsymbol{b} = \begin{bmatrix} 5 \\ 2 \end{bmatrix}$$



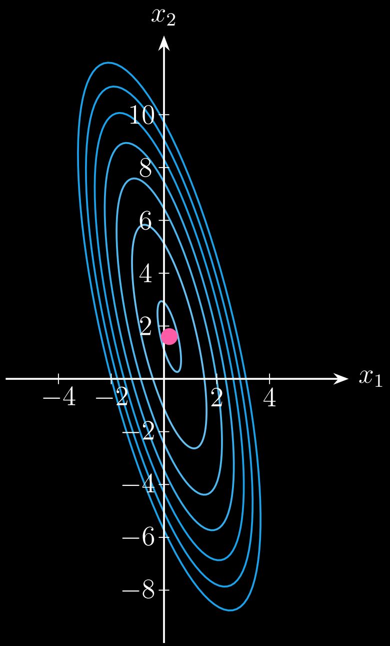
Quadratic optimization:

$$\min_{\boldsymbol{x}} \quad \frac{1}{2}\boldsymbol{x}^\top \boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}^\top \boldsymbol{x}$$

with

$$\boldsymbol{A} = \begin{bmatrix} 9 & 2 \\ 2 & 1 \end{bmatrix} \quad \text{and} \quad \boldsymbol{b} = \begin{bmatrix} 5 \\ 2 \end{bmatrix}$$

Optimal solution $\boldsymbol{x}^\star = \begin{bmatrix} 1/5 \\ 8/5 \end{bmatrix}$



Quadratic optimization:

$$\min_{\boldsymbol{x}} \quad \frac{1}{2}\boldsymbol{x}^\top \boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}^\top \boldsymbol{x}$$

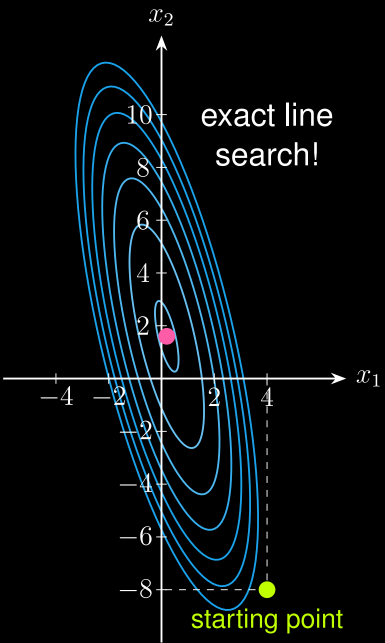
Steepest gradient descent:

$$\boldsymbol{g}_k = \boldsymbol{A}\boldsymbol{x}_k - \boldsymbol{b} \qquad \text{(gradient)}$$

$$\alpha_k = \frac{\boldsymbol{g}_k^\top \boldsymbol{g}_k}{\boldsymbol{g}_k^\top \boldsymbol{A}\boldsymbol{g}_k} \qquad \text{(step size)}$$

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \alpha_k \boldsymbol{g}_k \qquad \text{(decision vector)}$$

with an initial point $\boldsymbol{x}_0$.



Quadratic optimization:

$$\min_{\boldsymbol{x}} \quad \frac{1}{2}\boldsymbol{x}^\top \boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}^\top \boldsymbol{x}$$

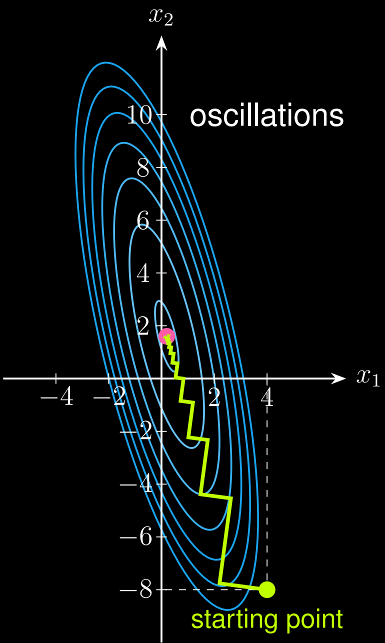
Steepest gradient descent:

$$\boldsymbol{g}_k = \boldsymbol{A}\boldsymbol{x}_k - \boldsymbol{b} \qquad \text{(gradient)}$$

$$\alpha_k = \frac{\boldsymbol{g}_k^\top \boldsymbol{g}_k}{\boldsymbol{g}_k^\top \boldsymbol{A}\boldsymbol{g}_k} \qquad \text{(step size)}$$

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \alpha_k \boldsymbol{g}_k \qquad \text{(decision vector)}$$

with an initial point $\boldsymbol{x}_0$.



Quadratic optimization:

$$\min_{\boldsymbol{x}} \quad \frac{1}{2}\boldsymbol{x}^\top \boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}^\top \boldsymbol{x}$$

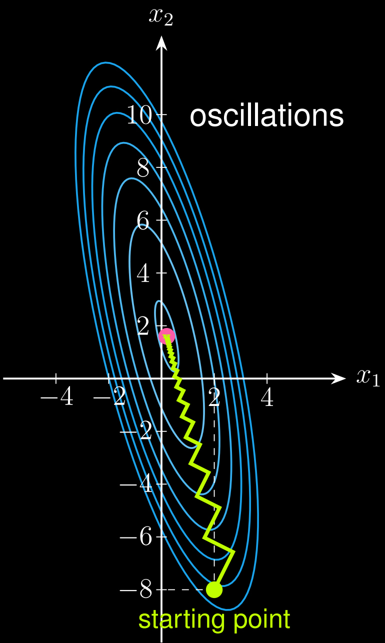
Steepest gradient descent:

$$\boldsymbol{g}_k = \boldsymbol{A}\boldsymbol{x}_k - \boldsymbol{b} \qquad \text{(gradient)}$$

$$\alpha_k = \frac{\boldsymbol{g}_k^\top \boldsymbol{g}_k}{\boldsymbol{g}_k^\top \boldsymbol{A}\boldsymbol{g}_k} \qquad \text{(step size)}$$

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \alpha_k \boldsymbol{g}_k \qquad \text{(decision vector)}$$

with an initial point $\boldsymbol{x}_0$.



```python
import numpy as np

# Problem parameters
A = np.array([[9, 2], [2, 1]])
b = np.array([5, 2])
x_star = np.linalg.solve(A, b)

# Steepest Gradient Descent implementation
x = np.array([2, -8])  # Starting point
path = [x.copy()]
max_iter = 150
for _ in range(max_iter):
    g = A @ x - b
    if np.linalg.norm(g) < 1e-6:
        break
    alpha = (g @ g) / (g @ A @ g)
    x = x - alpha * g
    path.append(x.copy())

path = np.array(path)
```

Positive Definite Matrix $\boldsymbol{A} = \begin{bmatrix} 9 & 2 \\ 2 & 1 \end{bmatrix}$

Eigenvalue decomposition:

$$\boldsymbol{A} = \lambda_1 \boldsymbol{u}_1 \boldsymbol{u}_1^\top + \lambda_2 \boldsymbol{u}_2 \boldsymbol{u}_2^\top$$

with eigenvalues:

$$\lambda_1 = 9.47 \quad \lambda_2 = 0.53$$

and condition number:

$$\kappa = \frac{\lambda_1}{\lambda_2} = 17.94$$

Positive Definite Matrix $\boldsymbol{A} = \begin{bmatrix} 9 & 2 \\ 2 & 1 \end{bmatrix}$

Eigenvalue decomposition:

$$\boldsymbol{A} = \lambda_1 \boldsymbol{u}_1 \boldsymbol{u}_1^\top + \lambda_2 \boldsymbol{u}_2 \boldsymbol{u}_2^\top$$

with eigenvalues:

$$\lambda_1 = 9.47 \quad \lambda_2 = 0.53$$

and condition number:

$$\kappa = \frac{\lambda_1}{\lambda_2} = 17.94$$

```python
import numpy as np

A = np.array([[9, 2], [2, 1]])
eig_val, eig_vec = np.linalg.eig(A)

print('lambda_1 = ', eig_val[0])
print('lambda_2 = ', eig_val[1])
print('u_1 = ', eig_vec[:, 0])
print('u_2 = ', eig_vec[:, 1])
print('condition number = ',
      eig_val.max() / eig_val.min())
```

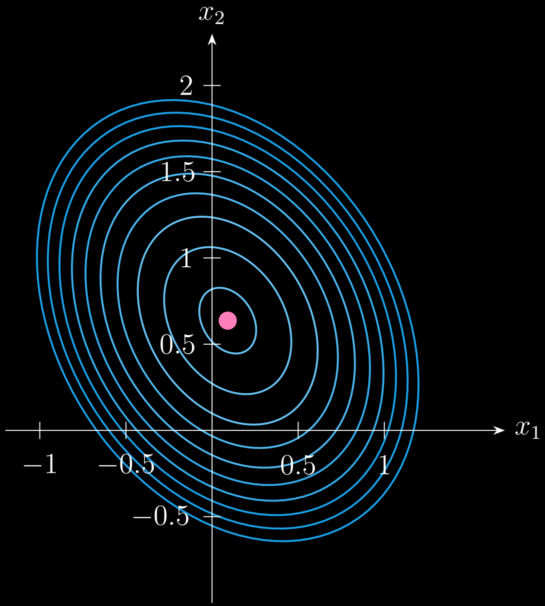
Quadratic optimization:

$$\min_{\boldsymbol{x}} \quad \frac{1}{2}\boldsymbol{x}^\top \boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}^\top \boldsymbol{x}$$

with

$$\boldsymbol{A} = \begin{bmatrix} 4 & 1 \\ 1 & 3 \end{bmatrix} \quad \text{and} \quad \boldsymbol{b} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$$

Optimal solution $\boldsymbol{x}^\star = \begin{bmatrix} 1/11 \\ 7/11 \end{bmatrix}$



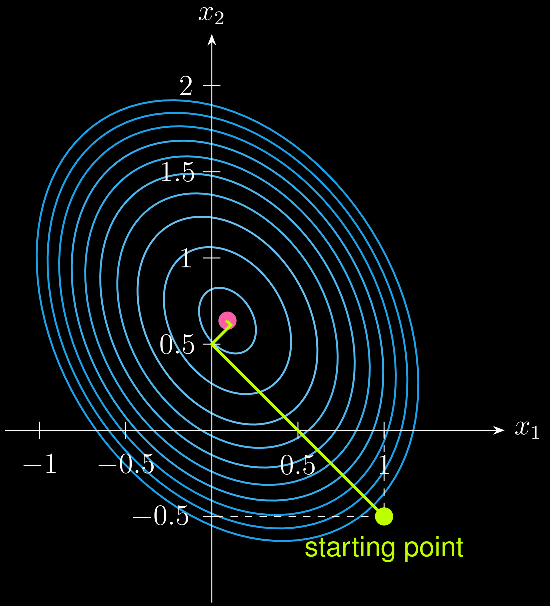
Quadratic optimization:

$$\min_{\boldsymbol{x}} \quad \frac{1}{2}\boldsymbol{x}^\top \boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}^\top \boldsymbol{x}$$

with

$$\boldsymbol{A} = \begin{bmatrix} 4 & 1 \\ 1 & 3 \end{bmatrix} \quad \text{and} \quad \boldsymbol{b} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$$

Optimal solution $x^\star = \begin{bmatrix} 1/11 \\ 7/11 \end{bmatrix}$



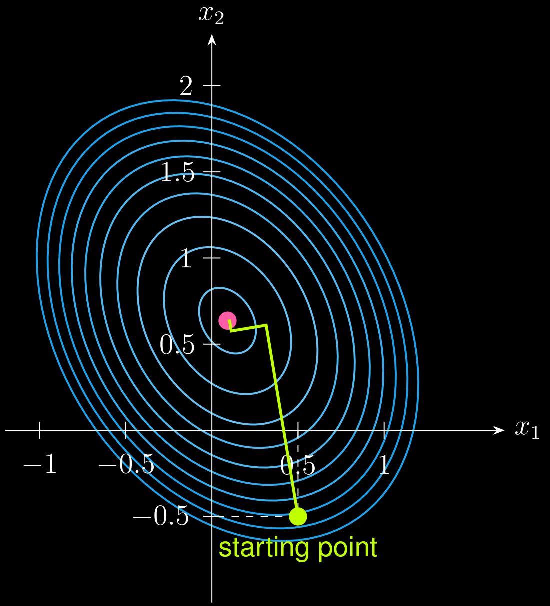
Quadratic optimization:

$$\min_{\boldsymbol{x}} \quad \frac{1}{2}\boldsymbol{x}^\top \boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}^\top \boldsymbol{x}$$

with

$$\boldsymbol{A} = \begin{bmatrix} 4 & 1 \\ 1 & 3 \end{bmatrix} \quad \text{and} \quad \boldsymbol{b} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$$

Optimal solution $x^\star = \begin{bmatrix} 1/11 \\ 7/11 \end{bmatrix}$



Positive Definite Matrix $\boldsymbol{A} = \begin{bmatrix} 4 & 1 \\ 1 & 3 \end{bmatrix}$

Eigenvalue decomposition:

$$\boldsymbol{A} = \lambda_1 \boldsymbol{u}_1 \boldsymbol{u}_1^\top + \lambda_2 \boldsymbol{u}_2 \boldsymbol{u}_2^\top$$

with eigenvalues:

$$\lambda_1 = 4.62 \qquad \lambda_2 = 2.38$$

and condition number:

$$\kappa = \frac{\lambda_1}{\lambda_2} = 1.94$$

```python
import numpy as np

A = np.array([[4, 1], [1, 3]])
eig_val, eig_vec = np.linalg.eig(A)

print('lambda_1 = ', eig_val[0])
print('lambda_2 = ', eig_val[1])
print('u_1 = ', eig_vec[:, 0])
print('u_2 = ', eig_vec[:, 1])
print('condition number = ',
      eig_val.max() / eig_val.min())
```

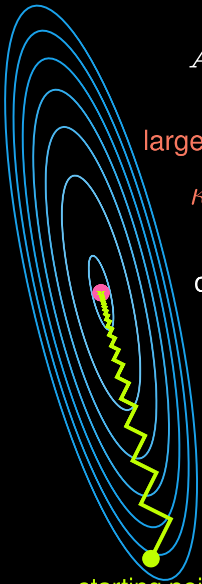
Steepest Gradient Descent

$$\boldsymbol{A} = \begin{bmatrix} 9 & 2 \\ 2 & 1 \end{bmatrix}$$

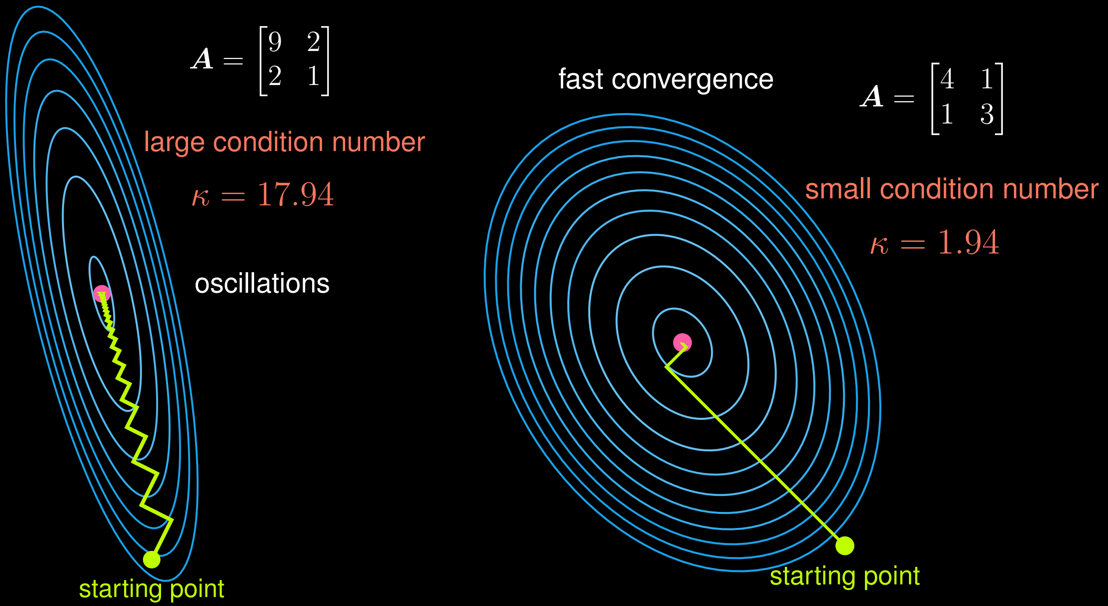
large condition number

$$\kappa = 17.94$$

oscillations



Steepest Gradient Descent



Convergence Rate of Steepest Gradient Descent

Quadratic optimization:

$$\min_{\boldsymbol{x}} \quad f(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^\top \boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}^\top \boldsymbol{x} \qquad \text{where} \qquad \boldsymbol{x}^\star = \boldsymbol{A}^{-1}\boldsymbol{b}$$

Linear convergence:

$$f(\boldsymbol{x}_{k+1}) - f(\boldsymbol{x}^\star) \leq \underbrace{\left(\frac{\kappa - 1}{\kappa + 1}\right)^2}_{\text{convergence rate } r} \left(f(\boldsymbol{x}_k) - f(\boldsymbol{x}^\star)\right)$$

Convergence Rate of Steepest Gradient Descent

Quadratic optimization:

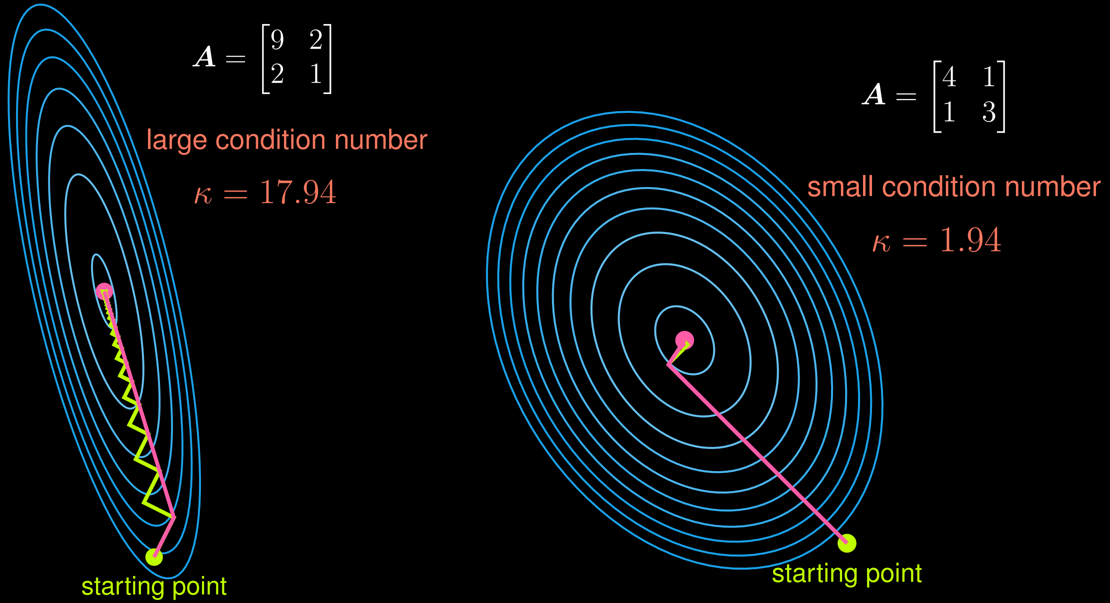
$$\min_{\boldsymbol{x}} \quad f(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^\top \boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}^\top \boldsymbol{x} \qquad \text{where} \qquad \boldsymbol{x}^\star = \boldsymbol{A}^{-1}\boldsymbol{b}$$

Linear convergence:

$$f(\boldsymbol{x}_{k+1}) - f(\boldsymbol{x}^\star) \leq \underbrace{\left(\frac{\kappa - 1}{\kappa + 1}\right)^2}_{\text{convergence rate } r} \left(f(\boldsymbol{x}_k) - f(\boldsymbol{x}^\star)\right)$$

- For $\boldsymbol{A} = \begin{bmatrix} 9 & 2 \\ 2 & 1 \end{bmatrix}$ ($\kappa = 17.94$) $\implies r = 0.8$ (slow convergence)
- For $\boldsymbol{A} = \begin{bmatrix} 4 & 1 \\ 1 & 3 \end{bmatrix}$ ($\kappa = 1.94$) $\implies r = 0.1$ (fast convergence)

Steepest Gradient vs. Conjugate Gradient



Thanks for your attention!

About me:

Homepage: https://xinychen.github.io

How to reach me: chenxy346@gmail.com