# Optimization Methods (CS1.404), Spring 2024
# Lecture 11


**Naresh Manwani**

Machine Learning Lab, IIIT-H


February 15th, 2024

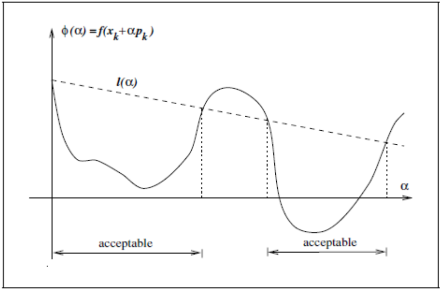
# Armijo Line Search Method

- Armijo inexact line search condition stipulates that $\alpha_k$ should first of all give sufficient decrease in the objective function $f$ , as measured by the following inequality:

$$f(\mathbf{x}_k + \alpha_k \mathbf{d}_k) \leq f(\mathbf{x}_k) + c_1 \alpha_k \nabla f(\mathbf{x}_k)^T \mathbf{d}_k$$

  for some constant $c_1 \in (0, 1)$.
- Thus, the reduction in $f$ should be proportional to both the step length $\alpha_k$ and the directional derivative $\nabla f(\mathbf{x}_k)\mathbf{d}_k$.

# Geometric Interpretation of Armijo Condition




- Consider $\phi(\alpha) = f(\mathbf{x}_k + \alpha \mathbf{d}_k)$ and $l(\alpha) = f(\mathbf{x}_k) + c_1 \alpha_k \nabla f(\mathbf{x}_k)^T \mathbf{d}_k$.
- The function $l(\alpha)$ has negative slope $c_1 \nabla f(\mathbf{x}_k)^T \mathbf{d}_k$, but because $c_1 \in (0,1)$, it lies above the graph of $\phi$ for small positive values of $\alpha$.
- The sufficient decrease condition states that $\alpha$ is acceptable only if $\phi(\alpha) \leq l(\alpha)$. In practice, $c_1$ is chosen to be quite small, say $c_1 = 10^{-4}$.

# Backtracking Line Search With Armijo

## Backtracking

1. **Initialize:** $\alpha^{(0)} \in (0,1)$, $\tau \in (0,1)$, $l = 0$
2. Until $f(\mathbf{x}_k + \alpha^{(l)}\mathbf{d}_k) > f(\mathbf{x}_k) + c_1\alpha^{(l)}\nabla f(\mathbf{x}_k)^T\mathbf{d}_k$
    1. Set $\alpha^{(l+1)} = \tau\alpha^{(l)}$
    2. $l = l + 1$
3. $\alpha_k = \alpha^{(l)}$

In practice the following choices are used

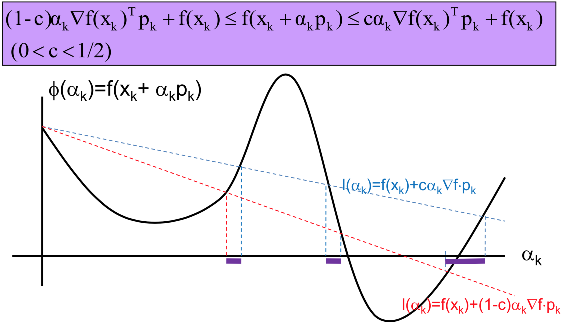
- $\tau \in (0.1, 0.5]$
- $c_1 \in [10^{-5}, 10^{-1}]$

# Issue with Armijo's condition:

- It does not ensure that the step size is sufficiently large because Armijo's condition can be satisfied even with a very small step size.
- Backtracking partially addresses this by starting from large step-sizes and checking Armijo condition.
- But is there some other condition that we can add to Armijo?

- Armijo-Goldstein inexact line search condition requires that $\alpha_k$ should be sufficiently large and it should give sufficient decrease in the objective function $f$ as well.
- The condition is as follows.
  $$f(\mathbf{x}_k) + (1 - c_1)\alpha_k \nabla f(\mathbf{x}_k)^T \mathbf{d}_k \leq f(\mathbf{x}_k + \alpha_k \mathbf{d}_k) \leq f(\mathbf{x}_k) + c_1 \alpha_k \nabla f(\mathbf{x}_k)^T \mathbf{d}_k$$
  for some constant $c_1 \in (0, 1/2)$.
- The first inequality is introduced to control the step length from below.
- **Issue:** First inequality may exclude all minimizers of $\phi$ (see in figure). One can see that the Goldstein condition misses the first local minima.

Geometrical Interpretation of Goldstein Conditions



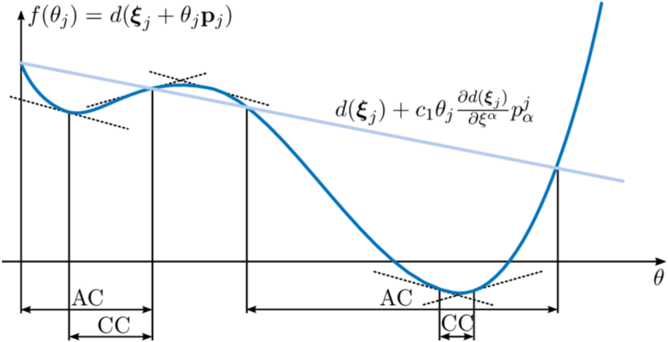
# Armijo-Wolfe Condition

- Armijo-Wolfe condition is also used to rule out unacceptably short steps (called the curvature condition) and ensure sufficient decrease.
- The conditions are

$$f(\mathbf{x}_k + \alpha_k \mathbf{d}_k) \leq f(\mathbf{x}_k) + c_1 \alpha_k \nabla f(\mathbf{x}_k)^T \mathbf{d}_k$$
$$\nabla f(\mathbf{x}_k + \alpha_k \mathbf{d}_k)^T \mathbf{d}_k \geq c_2 \nabla f(\mathbf{x}_k)^T \mathbf{d}_k$$

  for some constants $0 < c_2 < c_1 < 1$.
- LHS in the curvature condition is simply the derivative $\phi'(\alpha_k)$. So, the curvature condition ensures that the slope of $\phi$ at $\alpha_k$ is greater than $c_2$ times the initial slope $\phi'(0)$.
- If the slope $\phi'(\alpha)$ is strongly negative, we have an indication that we can reduce $f$ significantly by moving further along the chosen direction. if $\phi'(\alpha_k)$ is only slightly negative or even positive, then we cannot expect more decrease in $f$ in this direction, so it makes sense to terminate the line search.
- Thus, Wolf condition ensures sufficient rate of decrease of function value in the given direction.
- **Issue:** A step length may satisfy the Armijo-Wolfe conditions without being particularly close to a minimizer of $\phi$.

# Armijo-Wolfe Condition



# Exact Line Search for Quadratic Function

**Result**

- Let $f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T A\mathbf{x} + \mathbf{b}^T\mathbf{x} + c$, where $A$ is an $n \times n$ symmetric positive definite matrix, $\mathbf{b} \in \mathbb{R}^n$ and $c \in \mathbb{R}$.
- Let $\mathbf{x} \in \mathbb{R}^n$ and $\mathbf{d} \in \mathbb{R}^n$ be a descent direction of $f$ at $\mathbf{x}$.

Then

$$\arg\min_{t \geq 0} \; f(\mathbf{x} + t\mathbf{d}) = -\frac{\nabla f(\mathbf{x})^T \mathbf{d}}{\mathbf{d}^T A\mathbf{d}}$$

# Steepest Gradient Descent

- In the gradient method, the descent direction is chosen as the negative of the gradient at the current point: $\mathbf{d}_k = -\nabla f(\mathbf{x}_k)$. For such $\mathbf{d}_k$, we see that $\nabla f(\mathbf{x}_k)^T \mathbf{d}_k = -\|\mathbf{d}_k\|^2 < 0$.
- This is also the steepest gradient descent direction.

**Lemma**

Let $f$ be a continuously differentiable function, and let $\mathbf{x} \in \mathbb{R}^n$ be a non-stationary point (i.e., $\nabla f(\mathbf{x}) \neq \mathbf{0}$). Then the optimal solution of

$$\min_{\mathbf{d} \in \mathbb{R}^n} \nabla f(\mathbf{x})^T \mathbf{d}$$

$$\text{s.t. } \|\mathbf{d}\| = 1$$

is $\mathbf{d} = -\frac{\nabla f(\mathbf{x})}{\|\nabla f(\mathbf{x})\|}$.

# Steepest Gradient Descent Algorithm

- **Input:** $\epsilon > 0$ - tolerance parameter
- **Initialization:** Pick $\mathbf{x}_0 \in \mathbb{R}^n$ arbitrarily.
- **General Step:** For any $k = 0, 1, 2, \ldots$ execute the following steps
  1. Fix $\mathbf{d}_k = -\nabla f(\mathbf{x}_k)$
  2. Pick stepsize $t_k$ by a line search on the function

$$g(t) = f(\mathbf{x}_k + t\mathbf{d}_k)$$

  3. Set $\mathbf{x}_{k+1} = \mathbf{x}_k + t_k\mathbf{d}_k$
  4. If $\|\nabla f(\mathbf{x}_k)\| \leq \epsilon$, then stop and $\mathbf{x}_{k+1}$ is the output.

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY
HYDERABAD

# Example 1: Gradient Descent with Exact Line Search on Quadratic Function

- Consider function $f(x, y) = x^2 + 2y^2$, whose optimal solution is $(0, 0)$ with optimal value 0.
- Let $(x_0, y_0) = (2, 1)$, $\epsilon = 10^{-5}$.
- The Gradient descent approach stops in 13 iterations and finds a solution which is pretty close to the optimal value. $(x^*, y^*) = (0.1254 * 10^{-5}, -0627 * 10^{-5})$.

```
iter_number =   1 norm_grad = 1.885618 fun_val = 0.666667
iter_number =   2 norm_grad = 0.628539 fun_val = 0.074074
iter_number =   3 norm_grad = 0.209513 fun_val = 0.008230
iter_number =   4 norm_grad = 0.069838 fun_val = 0.000914
iter_number =   5 norm_grad = 0.023279 fun_val = 0.000102
iter_number =   6 norm_grad = 0.007760 fun_val = 0.000011
iter_number =   7 norm_grad = 0.002587 fun_val = 0.000001
iter_number =   8 norm_grad = 0.000862 fun_val = 0.000000
iter_number =   9 norm_grad = 0.000287 fun_val = 0.000000
iter_number =  10 norm_grad = 0.000096 fun_val = 0.000000
iter_number =  11 norm_grad = 0.000032 fun_val = 0.000000
iter_number =  12 norm_grad = 0.000011 fun_val = 0.000000
iter_number =  13 norm_grad = 0.000004 fun_val = 0.000000
```

# Example 1: Gradient Descent with Constant Step Size on Quadratic Function

- Consider function $f(x, y) = x^2 + 2y^2$, whose optimal solution is $(0, 0)$ with optimal value 0.
- Let $(x_0, y_0) = (2, 1)$, $\epsilon = 10^{-5}$, $t_k = 0.1$.
- The Gradient descent approach stops in 58 iterations.
- The stepsize was too small which causes slow convergence.

```
iter_number =   1 norm_grad = 4.000000 fun_val = 3.280000
iter_number =   2 norm_grad = 2.937210 fun_val = 1.897600
iter_number =   3 norm_grad = 2.222791 fun_val = 1.141888
     :                  :                    :
iter_number =  56 norm_grad = 0.000015 fun_val = 0.000000
iter_number =  57 norm_grad = 0.000012 fun_val = 0.000000
iter_number =  58 norm_grad = 0.000010 fun_val = 0.000000
```

# Example 1: Gradient Descent with Backtracking Line Search on Quadratic Function

- Consider function $f(x, y) = x^2 + 2y^2$, whose optimal solution is $(0, 0)$ with optimal value 0.
- Let $(x_0, y_0) = (2, 1)$, $\epsilon = 10^{-5}$, $\tau = 0.5$, $s = 2$, $c_1 = 0.25$.
- The Gradient descent approach stops in 2 iterations and outputs exact optimal solution.
- **For this example, inexact line search performs better than exact line search.**

```
iter_number =    1 norm_grad = 2.000000 fun_val = 1.000000
iter_number =    2 norm_grad = 0.000000 fun_val = 0.000000
```

# Example 2: Gradient Descent with Backtracking Line Search on Quadratic Function

- Consider function $f(x, y) = x^2 + \frac{1}{100}y^2$, whose optimal solution is $(0, 0)$ with optimal value 0.
- Let $(x_0, y_0) = (\frac{1}{100}, 1)$, $\epsilon = 10^{-5}$, $\tau = 0.5$, $s = 2$, $c_1 = 0.25$.
- The Gradient descent approach stops in 201 iterations.

```
iter_number =   1 norm_grad = 0.028003 fun_val = 0.009704
iter_number =   2 norm_grad = 0.027730 fun_val = 0.009324
iter_number =   3 norm_grad = 0.027465 fun_val = 0.008958
      :                   :                    :
iter_number = 201 norm_grad = 0.000010 fun_val = 0.000000
```

# Convergence of Steepest Gradient Descent

- For different quadratic functions, we observe that the convergence time varies for gradient descent.
- Can we find a measure which can predict how many iterations are needed for convergence of Gradient method.
- This measure would quantify in some sense the hardness of the problem.
- One such measure which can partially answer the above question is **condition number**.

# Convergence of Steepest Gradient Descent with Exact Line Search for Quadratic Function

- Let $f(\mathbf{x}) = \mathbf{x}^T A\mathbf{x}$, where $A$ is a symmetric positive definite matrix.
- For Steepest descent, $\mathbf{d}_k = -\nabla f(\mathbf{x}_k) = -2A\mathbf{x}_k$.
- Exact line search will result in $t_k = \arg\min_{t\geq 0}\ f(\mathbf{x}_k + t\mathbf{d}_k) = \frac{\mathbf{d}_k^T\mathbf{d}_k}{2\mathbf{d}_k^T A\mathbf{d}_k}$. Using this, we get

$$
\begin{aligned}
f(\mathbf{x}_k + t_k\mathbf{d}_k) &= f(\mathbf{x}_k) + t_k^2\mathbf{d}_k^T A\mathbf{d}_k + 2t_k\mathbf{d}_k^T A\mathbf{x}_k \\
&= \mathbf{x}_k^T A\mathbf{x}_k + \frac{(\mathbf{d}_k^T\mathbf{d}_k)^2}{4\mathbf{d}_k^T A\mathbf{d}_k} + \frac{\mathbf{d}_k^T\mathbf{d}_k}{2\mathbf{d}_k^T A\mathbf{d}_k}\mathbf{d}_k^T(-\mathbf{d}_k) \\
&= \mathbf{x}_k^T A\mathbf{x}_k - \frac{1}{4}\frac{(\mathbf{d}_k^T\mathbf{d}_k)^2}{\mathbf{d}_k^T A\mathbf{d}_k} = \mathbf{x}_k^T A\mathbf{x}_k\left(1 - \frac{1}{4}\frac{(\mathbf{d}_k^T\mathbf{d}_k)^2}{(\mathbf{d}_k^T A\mathbf{d}_k)(\mathbf{x}_k^T A\mathbf{x}_k)}\right) \\
&= \mathbf{x}_k^T A\mathbf{x}_k\left(1 - \frac{1}{4}\frac{(\mathbf{d}_k^T\mathbf{d}_k)^2}{(\mathbf{d}_k^T A\mathbf{d}_k)(\mathbf{x}_k^T AA^{-1}A\mathbf{x}_k)}\right) \\
&= \left(1 - \frac{(\mathbf{d}_k^T\mathbf{d}_k)^2}{(\mathbf{d}_k^T A\mathbf{d}_k)(\mathbf{d}_k^T A^{-1}\mathbf{d}_k)}\right) f(\mathbf{x}_k)
\end{aligned}
$$

# Convergence of Steepest Gradient Descent with Exact Line Search for Quadratic Function

## Kantorovich Inequality

Let $A$ be a positive definite $n \times n$ matrix. Then for any $\mathbf{x} \in \mathbb{R}^n$ ($\mathbf{x} \neq \mathbf{0}$), the inequality

$$\frac{(\mathbf{x}^T\mathbf{x})^2}{(\mathbf{x}^T A\mathbf{x})(\mathbf{x}^T A^{-1}\mathbf{x})} \geq \frac{4\lambda_{max}(A)\lambda_{min}(A)}{(\lambda_{max}(A) + \lambda_{min}(A))^2}$$

holds.