Optimization Algorithms: Theory and Software Implementation

Prof. Thirumulanathan D

Department of Mathematics

Institute of IIT Kanpur

Lecture: 13

Hello everyone. This is the third lecture of week three. Recall that we are looking at optimization algorithms. In the first lecture of this week, we introduced the general structure of an optimization algorithm where in each iteration we must find a descent direction and a step size. In the previous lecture, we introduced two methods: the exact line search algorithm and the backtracking line search algorithm.
Note $x_k=x^k$, $d_k=d^k$, $\alpha_k= \alpha^k$
We recall the backtracking line search algorithm, which uses Armijo's condition:
$f(x_k + \alpha_k d_k) - f(x_k) \leq c_1 \alpha_k \nabla f(x_k)^T d_k$

The algorithm involves choosing a value of α that satisfies Armijo's condition. We first initialize values for α, ρ, and c_1. We then repeatedly reduce α by a factor of ρ until Armijo's condition is met.

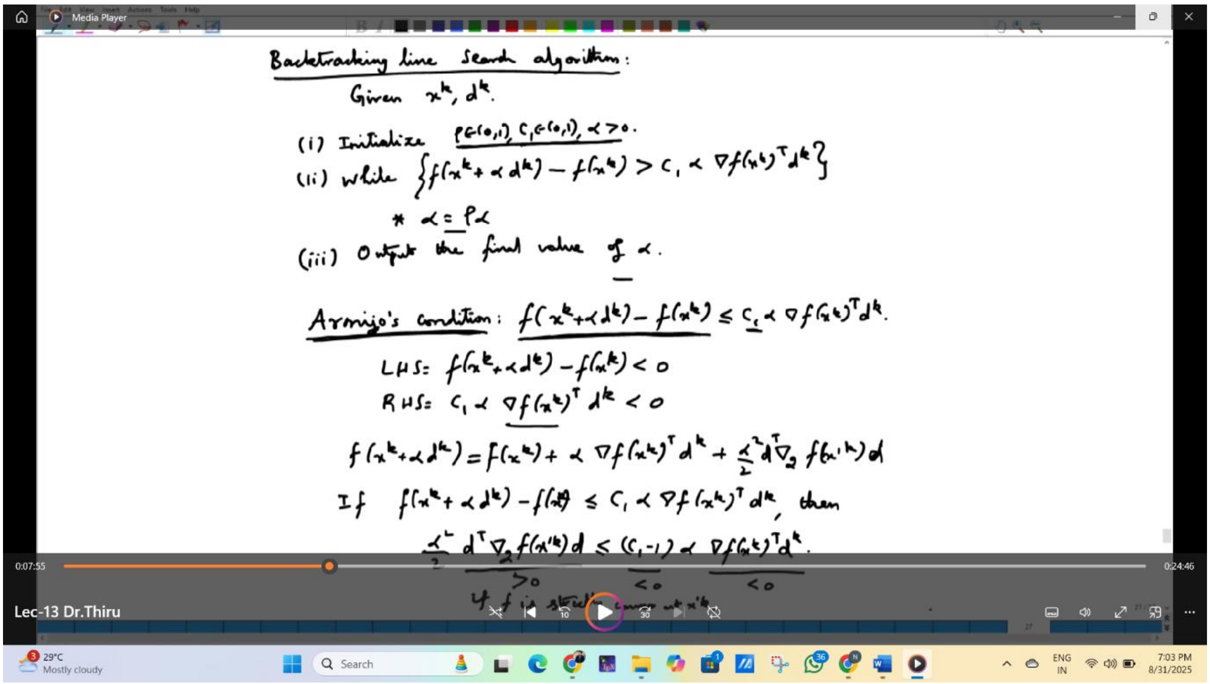
A natural question is why Armijo's condition works and what it ensures. To understand this, we start with a high value of α and reduce it step by step. For instance, starting with $\alpha = 1$, we might reduce it to 0.9, then 0.81, and so on. The purpose of reducing α is to ensure the condition is satisfied.

Let us examine Armijo's condition more closely.
The left-hand side, $f(x_k + \alpha d_k) - f(x_k)$, should be negative.
The right-hand side, $c_1 \alpha \nabla f(x_k)^T d_k$, is also negative because $\nabla f(x_k)^T d_k$ is negative (since d_k is a descent direction) and both c_1 and α are positive.
Armijo's condition requires that the decrease in the function value is not just negative but also sufficiently large relative to the gradient direction.

Expanding $f(x_k + \alpha d_k)$ using a Taylor series gives:
$f(x_k) + \alpha \nabla f(x_k)^T d_k + (\alpha^2/2) d_k^T \nabla^2 f(\bar{x}) d_k$
where $\bar{x}$ lies between x_k and $x_k + \alpha d_k$.

Substituting into Armijo's condition yields:
$(\alpha^2/2) d_k^T \nabla^2 f(\bar{x}) d_k \leq (c_1 - 1) \alpha \nabla f(x_k)^T d_k$

(Refer Slide Time 7:55)



The right-hand side is positive because (c_1 - 1) is negative and $\nabla f(x_k)^T d_k$ is negative. If the Hessian $\nabla^2 f(\bar{x})$ is positive definite, then $d_k^T \nabla^2 f(\bar{x}) d_k$ is positive.

This implies that α must be less than or equal to a certain value for the inequality to hold. If Armijo's condition is not satisfied for a given α, we reduce α until it is. This is the rationale behind Armijo's condition, proposed by Larry Armijo in 1966.

In the backtracking algorithm, we start with an initial α (e.g., α = 1). If Armijo's condition is not met, we multiply α by ρ (e.g., ρ = 0.8) and check again. This process continues until the condition is satisfied.

Let us illustrate with an example.
Consider $f(x) = x_1^2 + x_2^2$, with $x_0 = (5, 4)$ and $d_0 = (-1, 1)$.
Choose ρ = 0.8, $c_1 = 0.75$, and initial α = 0.5.

We compute the following values:

For α = 0.5:
$f(x_0 + \alpha d_0) - f(x_0) = -0.5$
$c_1 \alpha \nabla f(x_0)^T d_0 = -0.75$
Since -0.5 > -0.75, the condition is not satisfied. We set α = 0.4.

For α = 0.4:
$f(x_0 + \alpha d_0) - f(x_0) = -0.48$
$c_1 \alpha \nabla f(x_0)^T d_0 = -0.6$
Since -0.48 > -0.6, the condition is not satisfied.
We set α = 0.32.

For $\alpha = 0.32$:
$f(x_0 + \alpha d_0) - f(x_0) = -0.4352$
$c_1 \alpha \nabla f(x_0)^T d_0 = -0.48$
Since $-0.4352 > -0.48$, the condition is not satisfied.
We set $\alpha = 0.256$.

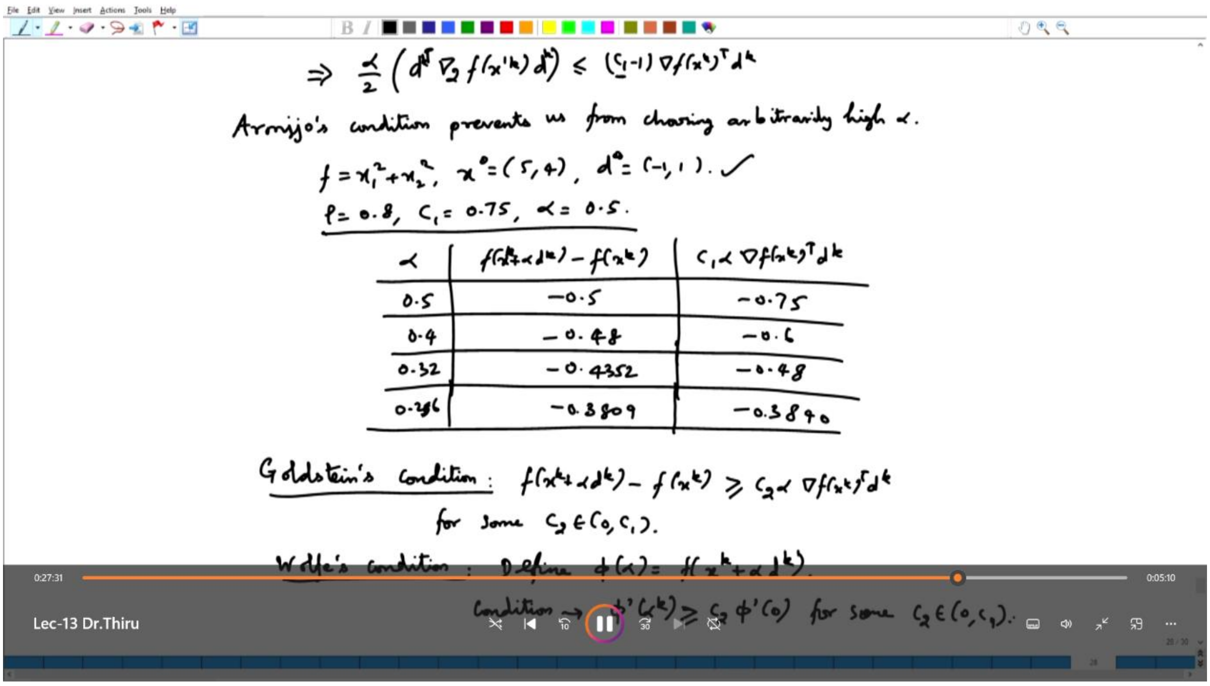
For $\alpha = 0.256$:
$f(x_0 + \alpha d_0) - f(x_0) = -0.3809$
$c_1 \alpha \nabla f(x_0)^T d_0 = -0.384$
Now $-0.3809 \leq -0.384$, so the condition is satisfied.
The algorithm returns $\alpha = 0.256$.

Note that the exact line search for this example gave $\alpha = 0.5$.
Backtracking line search provides an inexact value, but it is sufficient for convergence. Inexact line search methods, like backtracking, are widely applicable because they do not require solving for the exact minimum and can be used for any differentiable function.

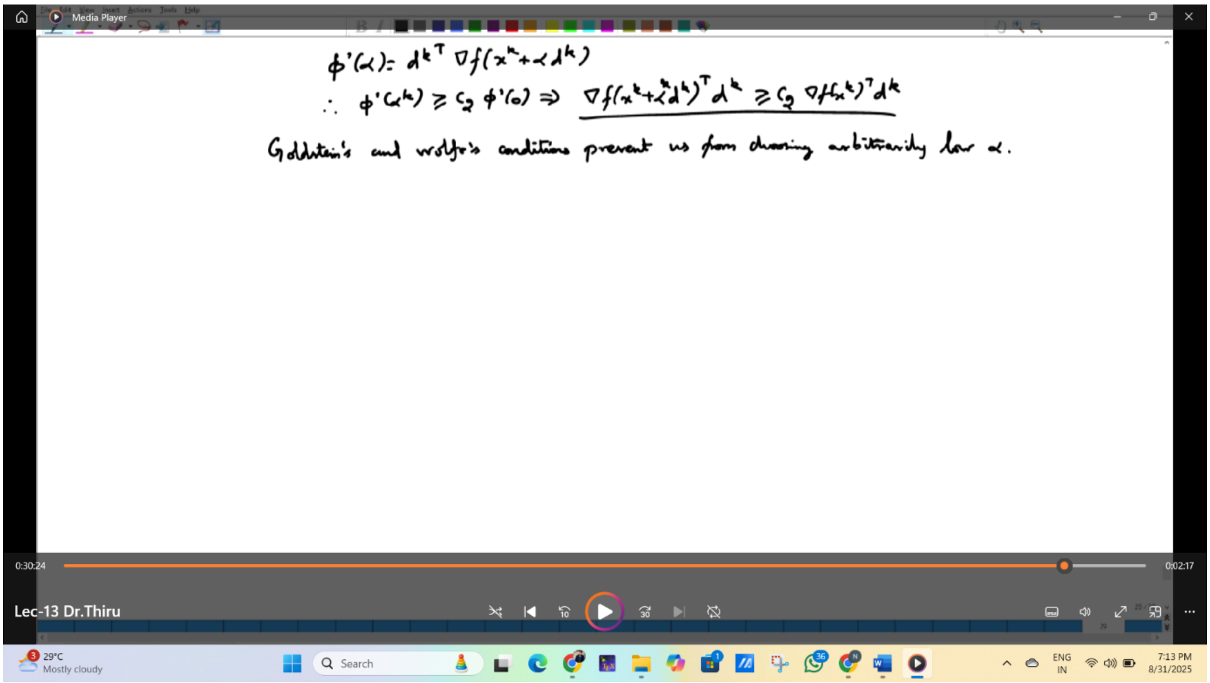
Armijo's condition prevents excessively large α but does not prevent very small α. To address this, we have Goldstein's and Wolfe's conditions.

Goldstein's condition requires:
$f(x_k + \alpha d_k) - f(x_k) \geq c_2 \alpha \nabla f(x_k)^T d_k$
for some c_2 in $(0, c_1)$. This ensures α is not too small.

(Refer Slide Time 27:05)



Wolfe's condition requires:
$\nabla f(x_k + \alpha_k d_k)^T d_k \geq c_2 \nabla f(x_k)^T d_k$
which similarly prevents overly small step sizes.
(Refer Slide Time 30:24)



In practice, backtracking line search with Armijo's condition is often sufficient, especially when ρ is chosen reasonably (e.g., $\rho \geq 0.5$), as it avoids extremely small α values.

This concludes our discussion on line search algorithms. In the next lecture, we will examine algorithms for choosing descent directions.

Thank you.