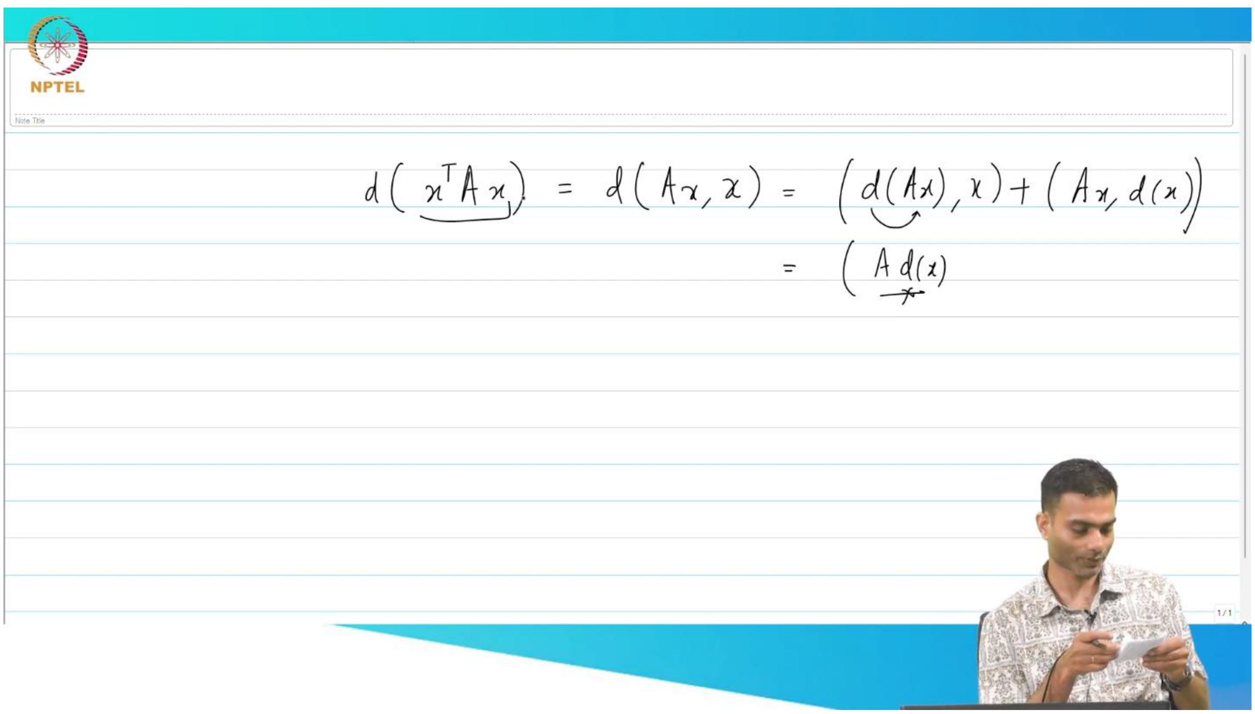
Course Name: Optimization Theory and Algorithms
Professor Name: Dr. Uday K. Khankhoje
Department Name: Electrical Engineering
Institute Name: Indian Institute of Technology Madras
Week - 03
Lecture - 15

Unconstrained Optimization -1- Roadmap of the Course and Taylor's Theorem

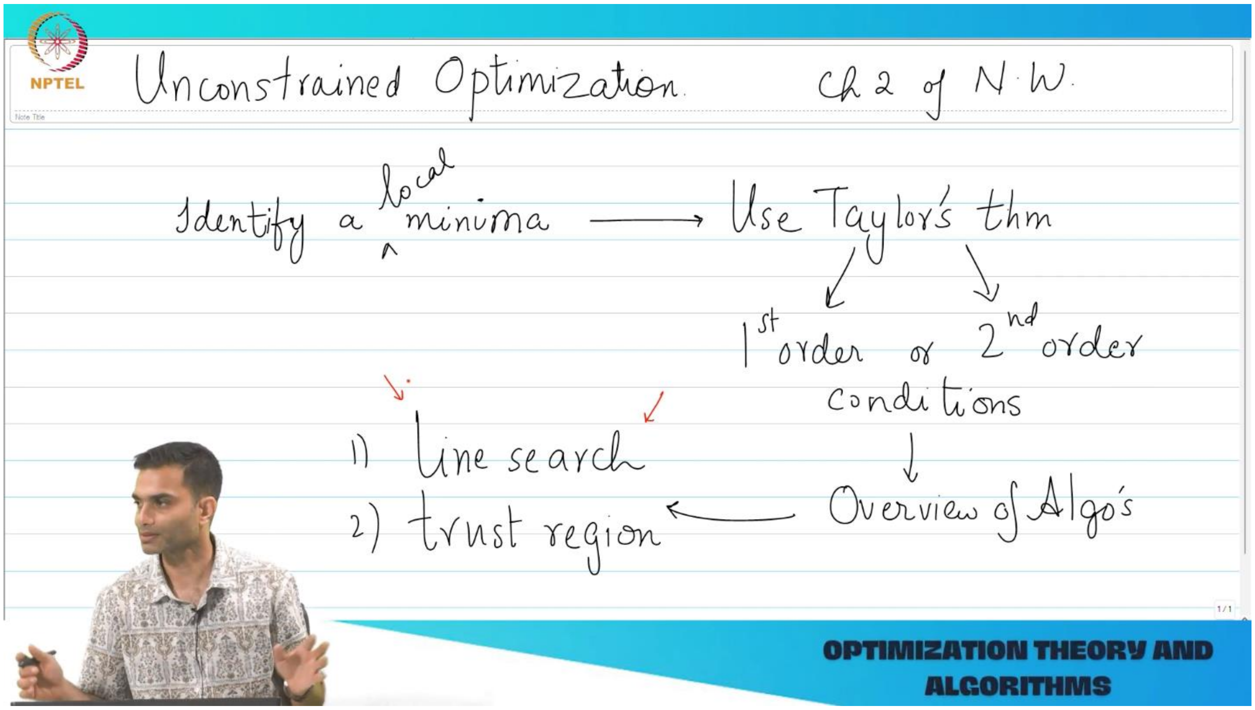
Okay, so let us get started with the doubt sheets. Someone has found a typo in the online notes. Thank you very much. I wrote ∇x instead of δx, which we wrote correctly during class yesterday, but in the online notes, it is ∇x. Anyone wants to claim this? No one? Okay. So, I had a prof in, not undergrad, during my PhD time; he was a Canadian prof.

And back then, to do our laundry, we had to use a coin-operated machine. So every time anyone would answer a correct question, he would throw a quarter to us and use that quarter for doing laundry. But I don't have any quarters for that. There's a question: can we apply the chain rule of differentiation to the inner product? I think the student was sitting right at the center last time.



That was you. Yeah, so what you have written is, let me just write that down. So, the question is can we do something like this $x^T Ax$ and he has written that as Ax. This is another way of writing an inner product, but the first step is kind of illegal. When you do this, you have said $d(Ax) = x + Ax\,dx$ which looks okay, but then the next step is you cannot move this d like this; what you have written is $A\,dx$. Now, because $A \cdot x$ scrambles up all the x's, you cannot move the derivative operator straight away inside; although you finally get the correct answer by doing that, this operation is illegal.

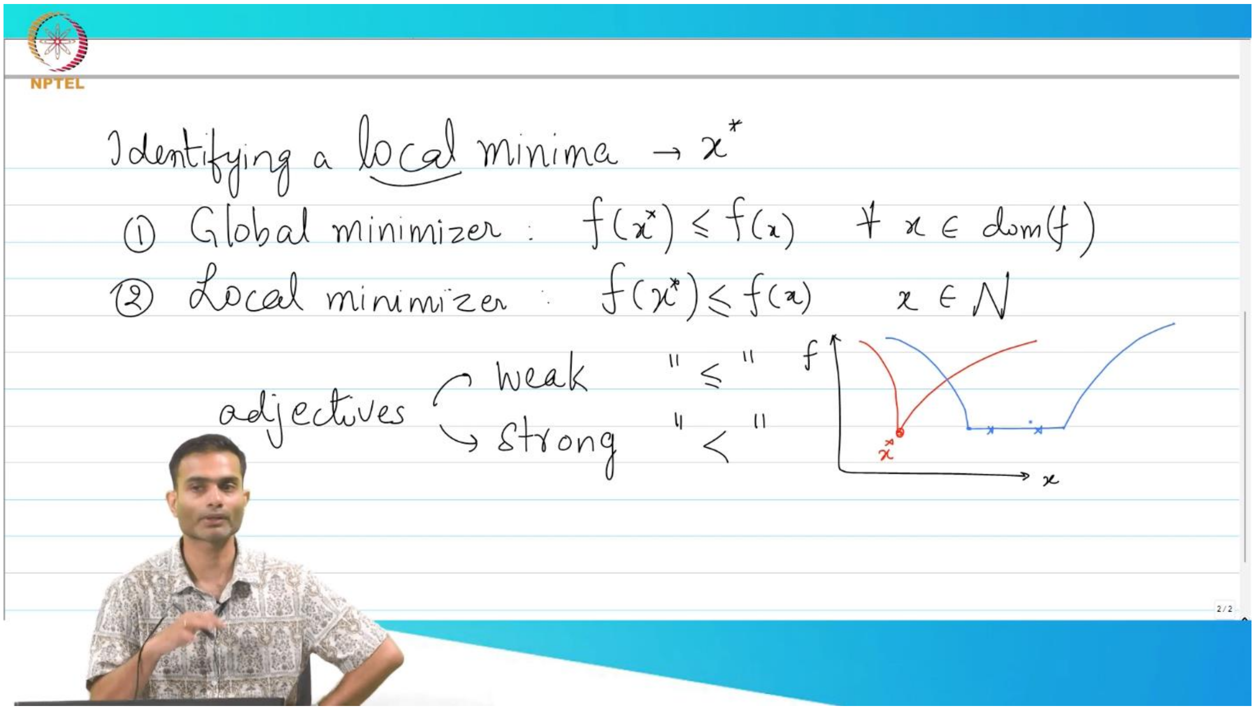
But you could come up with some equivalent way of writing it, but in our case, this is already an inner product. So, we do not have to write it separately like that, right? What are the applications of the Jacobian operation? Does it explain the relation between input and output variables? So, we saw that what was actually Jacobian expressing? Derivative. Derivative, but a little bit more. If you were to write it in plain English, what is the Jacobian expressing? The rate of change of each of the component functions with respect to each of the variables, right?



So, it is not exactly input-output; a better word would be the sensitivity of each function to each variable. Okay. We wrote the definition of the directional derivative for a scalar function $f(x)$, and the question is can we generalize it to a multivariable function? Yes, just go component-wise. Okay, multivariable calculus seems to be interesting, but also huge; I guess I need a little practice to get a good grasp. Can you provide some materials for practice? Has everyone seen this YouTube channel Three Blue One Brown, right? So, do they have stuff on multivariable calculus, or is it only linear algebra? I have seen linear algebra; I do not think I have seen multivariable calculus, but if anyone—this is open to everyone—those of you who have not seen Three Blue One Brown, please go and have a look at it; it is a great tool for linear algebra, and if someone finds a similar tool for calculus, that would be great. I do not know of a tool myself. One video. Oh, okay, nothing for multivariable. Okay, then this was Jaikar; is Jaikar here? Not here? Okay, the question was about the definition of ∇f and can you go both ways? It is not an "if and only if" statement that I have to go both ways; this definition so need not sweat it.

So, today's lecture, we will finally get into proper optimization, unconstrained optimization, right? The first half of the course is unconstrained; the second half is constrained. So, let us begin with unconstrained, okay? So, let me just give you a bit of an overview of what we will cover in this module. This is roughly chapter 2 of Nocedal and Wright, okay.

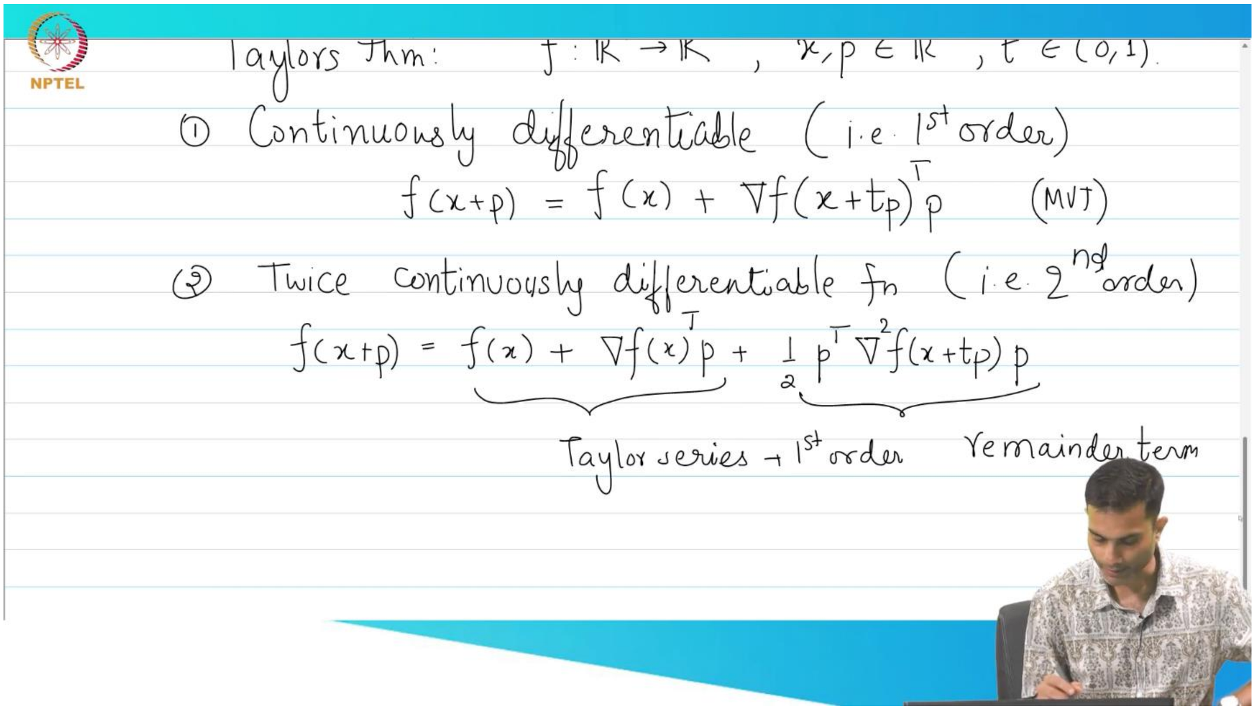
If you want to read the full textbook, you can read chapter 2, or you can read the online notes, okay? So, when we are talking about unconstrained optimization, or optimization in general, we could be talking about maximizing a function or minimizing a function. To avoid confusion, in general, I am going to talk about minimizing a function. If you have a maximization problem, all you need to do is multiply by -1; it becomes a minimization problem, okay? So, leaving aside all technical details, what is the nature of the game? Find a minima, right?



So, never lose sight of that; that is what we are trying to do, right? That is the bottom line. Now, the moment I write this, there is a little bit of ambiguity over here because a problem can have multiple minima, right? So, I should qualify this; I should say identify a local minima, okay? That is what I want to do. I want to identify a local minima. In general, finding a global minima can be very challenging; we will talk about it as we go, but something simpler is to identify a local minima.

So, what do I mean by identify? Some kind of a test that if I get a point, how do I verify that, you know, is it a candidate for a local minima? So, we want to look at tests for this, okay? There come—we have basically—we are going to talk about two different types of tests depending on how differentiable the function is. So, the building block for this is multivariate Taylor's theorem, okay? So, we are going to use Taylor's theorem, and depending on how differentiable the function is, I am going to either be able to use first-order or second-order, let us call them conditions.

Conditions for what? Conditions to identify the local minima, okay? From here, we will sort of give an overview of algorithms, okay? Again, in the world of algorithms, broadly there are two classes: one is what is called a line search; the other is called a trust region. Two different approaches for solving optimization problems, okay? In this course, we are primarily going to talk about line search algorithms, okay? That itself is extremely vast. So, this is just sort of the roadmap, okay?



So, our theme now is identifying; that is the first thing I want to do. Now, to give a special word for this local minima, I am going to write it as x^*. So, if you see x^*, in general, it denotes this point, which is a local minima, okay? And because I have already given you a hint that there may be multiple minima, there must be some hierarchy of minima, right? So, the hierarchies are local minima and the best of all minima—what would you call it? Global minima, right?

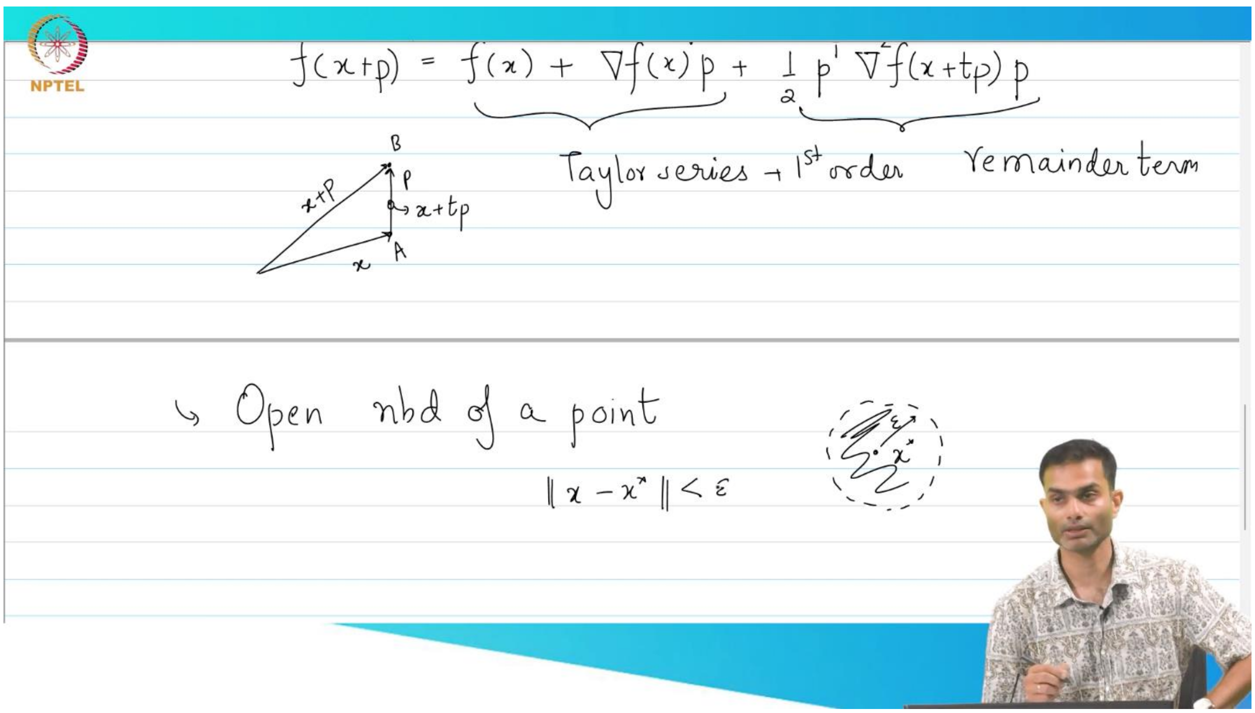
So, the global minimizer obviously satisfies $f(x^*) \leq f(x)$ for all x in the domain of f. This is the natural definition of the global minimizer. Then you come to the local minimizer, okay? Obviously, this statement would be the same $f(x^*) \leq f(x)$, but now x belongs to some set or neighborhood of some point, okay? You will find in the literature when people talk about minimizers, they add one more adjective; they may not just call it a minimizer; they may call it a weak minimizer or a strong minimizer. So, these are the adjectives.

These words are fairly self-explanatory. So, if you were in charge of definitions, weak would mean what? Okay, what would be a strong minima? So, it is a little different; if you see $\leq$, it is a weak minimizer; if you see $<$, it is a strong minimizer. So, I can have a weak global minimizer; I can have a strong global minimizer, right? And the best way to see it is let me plot for you: this is, let us say x; this is f, right? And I have a function that comes like this and goes like this, right? Now, this point over here x^* is clearly a strong minimizer; it is a strong global minimizer.

I could also have something like this, where it comes over here, right? Now, I have these all points on the same line; they are all minimizers. So, they are weak minimizers because that equality is being satisfied, and you can have both that are happening. So, this $<$ or $\leq$ can make a difference, okay? So, these subtle points can become hard to verify in multiple dimensions, right?

So, we have to, you know, usually we are kind of careless about $<$ or $\leq$, but this can make a big difference. So, if different people are solving the same problem from different starting points and they end up with different solutions, that is possible, right? One guy started and ended up here; another guy started and ended up here; they are all correct, right? Now, what we will see is that in the whole game of optimization, particularly when we are talking about nonlinear optimization, your starting point becomes important; where you start often determines where you end. So, that is why you can start from some point over here and maybe end up here, or you could start from some point over here and maybe end up somewhere here.

So, the fact that we are dealing with nonlinear optimization is the first complication that we have to deal with. Now, one quick thing before we start is to show this to you in real-time. I have written this all down, so you will not find it in your notes yet, but I will also show it to you now. So, let me write down the multivariate Taylor expansion: we wrote this for one variable. The Taylor expansion is what?



It is like saying if $f: R^n \to R$, and if I have a point $x_0 \in R^n$ and I want to find $f(x)$ near x_0, I can expand around that point; so $f(x)$ is equal to $f(x_0) + (x - x_0)^\top \nabla f(x_0) + \frac{1}{2}(x - x_0)^\top H(x_0)(x - x_0) + o(\| x - x_0 \|^2)$, where $H(x_0)$ is the Hessian.

So this is for functions $f: R^n \to R$. So, if I have the second-order function $f(x)$ being twice continuously differentiable, you can expand this about any point x_0 in its domain.