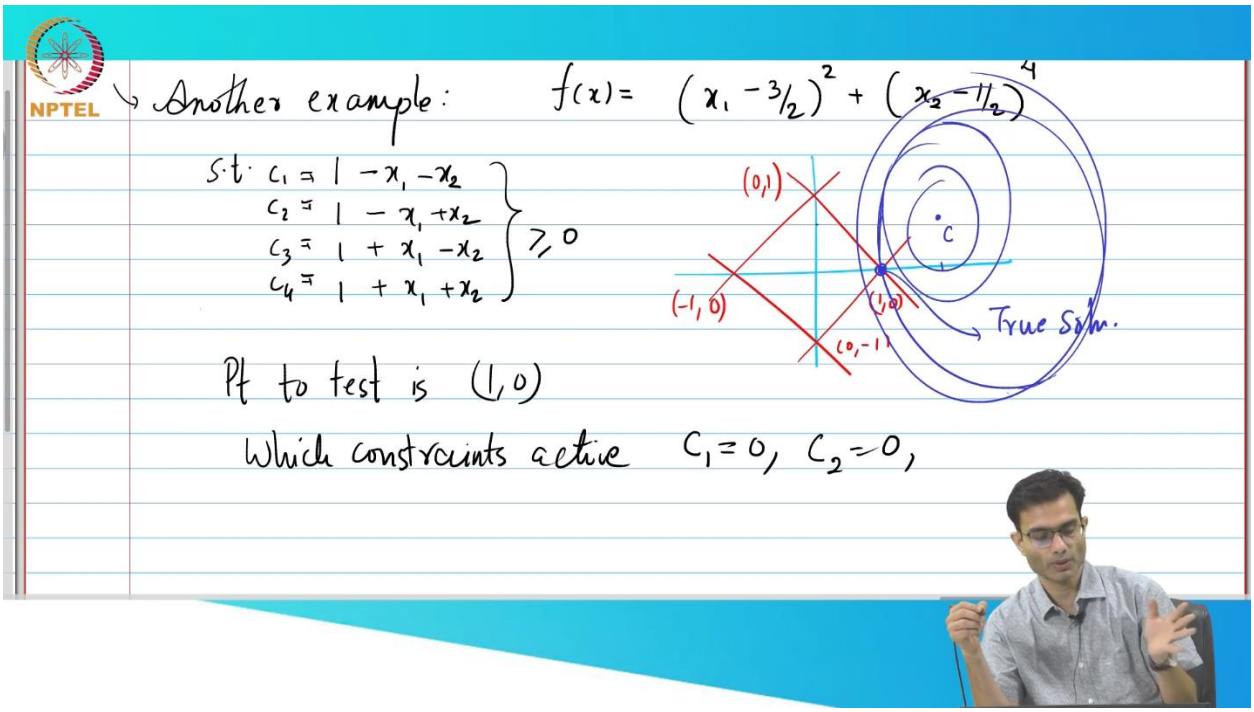
Course Name: Optimization Theory and Algorithms
Professor Name: Dr. Uday K. Khankhoje
Department Name: Electrical Engineering
Institute Name: Indian Institute of Technology Madras
Week - 10
Lecture - 71

Introduction to Projected gradient descent

Now let us take a totally different example. This time we are going to go to four constraints. Ok. So, but again geometry will help us solve this problem. Now, note I am going to pick a problem whose solution I can infer from geometry just so that it is easy to verify and understand. Ok. So, I am going to give you an objective function $f(x)$, this time it is not going to be simple linear objective $\left(x_1 - \frac{3}{2}\right)^2 + \left(x_2 - \frac{1}{2}\right)^{2^4}$ otherwise it would have been very simple to sketch. So, this is my objective function subject to:



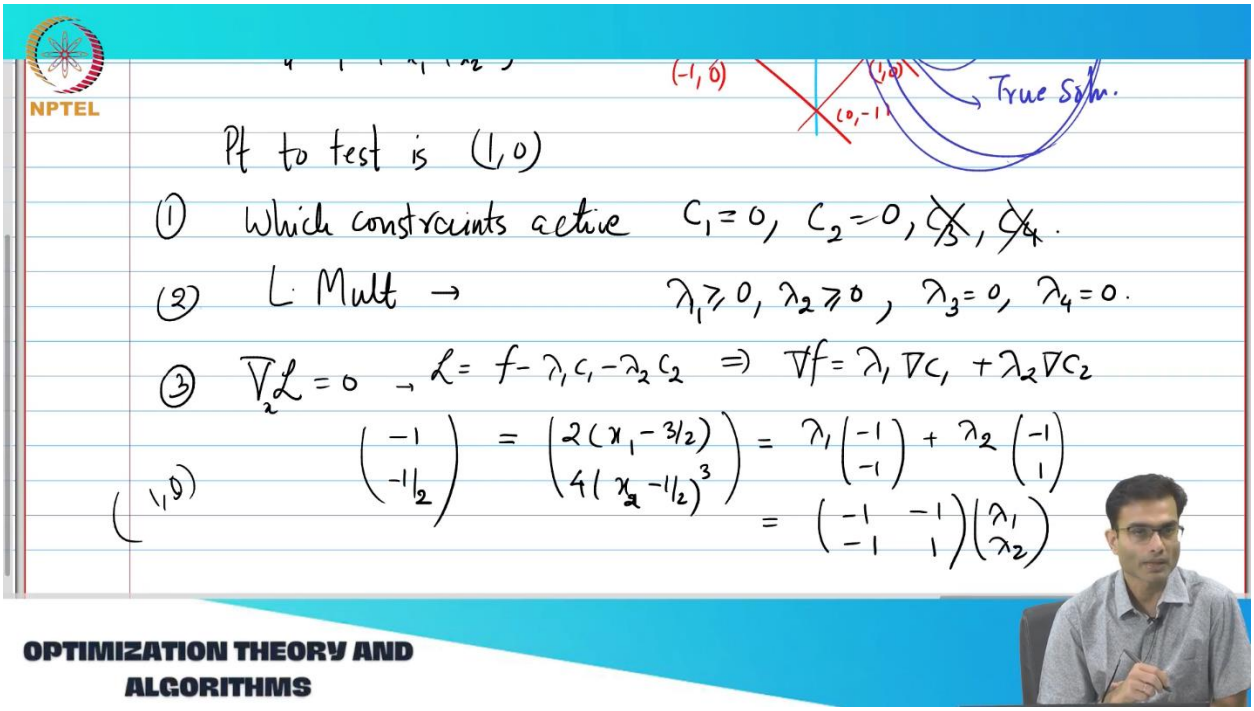
4 constraints, they look complicated, but actually they are not that complicated. First constraint: What does it look like? $1 \leq x_1 + x_2$, that means, it is a straight line with slope 135 degrees and I want to be below it, right? Second constraint: $1 \leq x_1 - x_2$, or you want to start with the fourth constraint. What is the fourth constraint? $x_1 + x_2 \geq -1$. So, it is a line parallel to this guy and I want to be above it, right?

So, it is giving me this. What about the other guys? So, it is actually a diamond. And very quickly you can confirm supposing I take the origin, the left-hand side in all four cases will become 1. Right? $1 \geq 0$. So, the origin belongs to this feasible set, which is what I mean that just when you draw a feasible set, it is good to take a few test points to make sure that you did not

flip an inequality and mess it up that way. Ok. So, this point is going to be what? The coordinate points are going to be $(1,1), (1,1)$, right?

So, the points are $(0,-1), (-1,0), (0,1)$. These are the 4 points, alright? Ok. Now, comes the part of the objective function. So, it is not a quadratic form, that much we can see; one is squared, and one is to the power 4. But we can identify the point at which this function goes to 0, which is $\left(\frac{3}{2}, \frac{1}{2}\right)$, right? So, $\frac{3}{2}$ here and $\frac{1}{2}$ maybe here, right? So, this is the point where this function goes to 0, right? And from here I can draw some contours. They would not be circles because it would be circles if it were squared instead of to the power of 4, but there will be contours that will look somewhat like this. Right?

We can later draw this on MATLAB or whatever and see. And so, you will just have to take my word on this that the contours will look like this. Ok. And the way these contours work is, you know, they would sort of go like this. So, you know, without having plotted this, if I had an actual MATLAB plot, you would see that the subsequent ones would go inside it. So, the contour which touches the blue contour, which touches the red line at the earliest, is the guy that is going to be the solution. As the contours grow, the function value is increasing, right? The least value of f is at the dot, right?



So, at the center C over here, as I start increasing the function values, these contours grow bigger and bigger and bigger, right? So, this point at which this guy touches over here is actually the solution to this problem because if I go any further into the feasible set, the function value is going to increase. Ok. So, you have to take my crappy drawing at face value for believing this. Ok. So, this happens to be the true solution. Ok. So, here is also how the KKT theorem helps us to check our intuition. Will it help us check our intuition? We will take this point $(1,0)$ and subject it to KKT. If it works, then it is a solution. Ok.

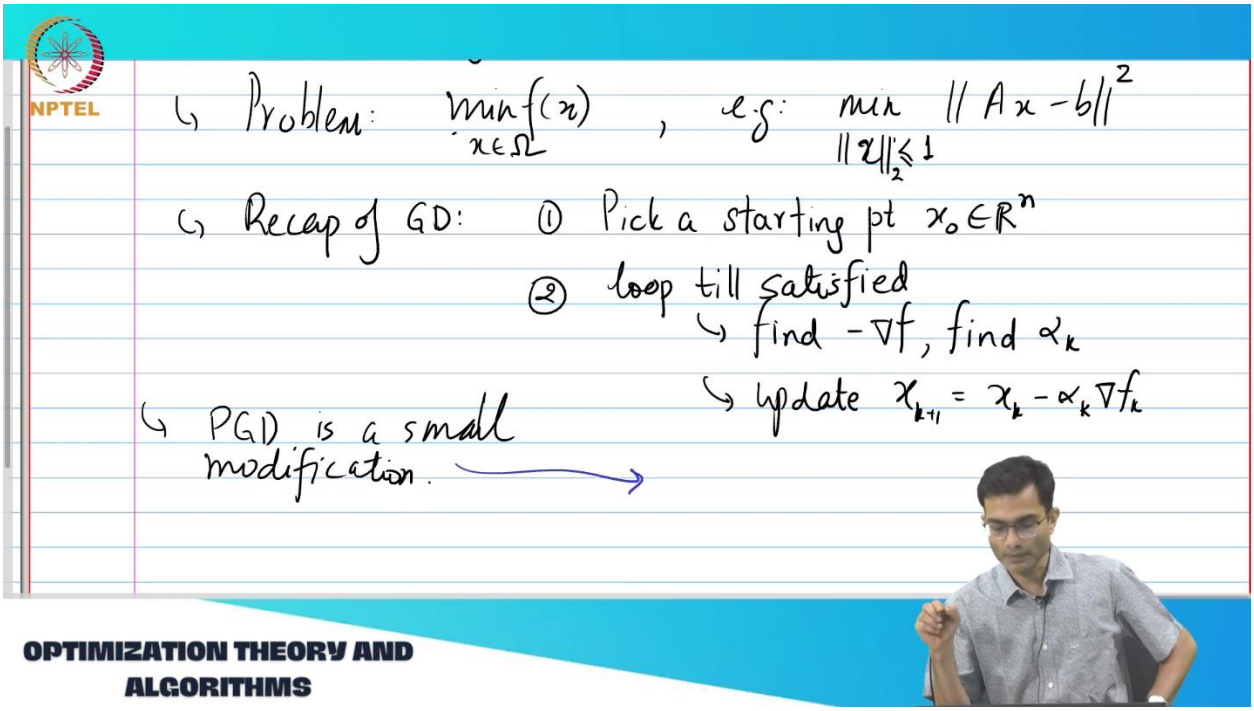
So, this is C_1, C_2, C_3, C_4. I should not put equal to, but yeah, it is good to say equal to. Alright. Now, at $(1,0)$, which constraints are active? C_1, right? You can substitute into this: $1 - 1 = 0$. First one $C_1 = 0$. C_2: If I substitute $1 - 1 + 0 = 0$. C_3: If I substitute $1 + 1 = 2 - 0$, so $2 > 0$.

So, not C_3. C_4: $1 + 1 = 2 > 0$. So, not C_3, right? So, I have two constraints that are active, right? So, when I write my Lagrangian multipliers, which ones do I have to set to 0? So, this is step 1. Step 2 is Lagrange multipliers.

Which one should I set to 0 straight away without doing any calculation? $\lambda_3 = 0, \lambda_4 = 0$. So, let us set those guys to 0 immediately. $\lambda_3 = 0, \lambda_4 = 0$, done. Can I say something about λ_1, λ_2? What? ≥ 0. Why? Inequality constraints. Ok. So, like this, we proceed systematically. Excellent. Next is, let us get this Lagrangian into business. If this guy $(1,0)$ is the optimal point, what should happen? $\nabla L = 0$. $\nabla L = 0$. So, let us look at ∇L.

So, first of all, what is L over here? $f - \lambda_1 C_1 - \lambda_2 C_2$. Do I need to write λ_3, λ_4? No, they are 0. So, this implies that $\nabla f = \lambda_1 \nabla C_1 + \lambda_2 \nabla C_2$. So, now let us write that down.

What is ∇f? $2\left(x_1 - \frac{3}{2}\right)$ (that is the first component), then $4\left(x_1 - \frac{1}{2}\right)^3$ (this is my ∇f). And what is ∇C_1? $(-1,-1)$. So, $\lambda_1(-1,-1)$, and what is ∇C_2? $(-1,1)$. Ok. The point is $(1,0)$. So, if I substitute $(1,0)$ into this, what will I get? $1 - \frac{3}{2} = -\frac{1}{2}$, multiplied by 2 gives -1. So, this is -1 and this guy is 0.



So, what do we get? Minus half because it is minus 1, 8 into 4, which is minus half. So, λ_1 and λ_2 have to satisfy this. Ok. Do you just by looking at it, do you think it is going to be possible? Yes. So, what was your intuitive way of arriving at it without solving? That is very important. They are linearly independent, right? I could write this as:

$$\begin{pmatrix} -1 & -1 \\ -1 & 1 \end{pmatrix} \begin{pmatrix} \lambda_1 \\ \lambda_2 \end{pmatrix}$$

I could write it like this also, and you can look at this matrix, right? It is invertible because the two columns are not linearly dependent. So, it should be possible to invert it. However, that is not good enough, right? What do we have to also check? We have to check whether it is positive, I mean non-negative or not. So, can you solve this? What do we get? $\lambda_1 = \frac{3}{4}$, that is right, and $\lambda_2 = \frac{1}{2}$, which is greater than or equal to 0. So, our test point $(1,0)$ turns out to be the optimum point. So, you do not have to have perfect drawing skills or MATLAB or quickly subject your KKT test and you are done. Ok. Right?

So, straightforward again just to revise what we did. Bring your candidate point, identify which constraints are active. Why do we do that? So, that is step 2. We can knock off the lambdas which are 0, right? So, $\lambda_3, \lambda_4 = 0$, we got rid of them. From that, we form the Lagrangian. Form the Lagrangian, we said $\nabla L = 0$ and then we solved for it. Is it possible? Answer is yes. Are those Lagrange multipliers non-negative for inequality constraints? Yes. The point passes the test. Ok. If I had an equality constraint in here, what is the thing that I have to not worry about? Non-negativity need not be the case. Ok.

So, that is what we are going to do as far as first-order necessary conditions for constraint optimization. Ok. There is a huge amount of problems that you can solve using this alone. Ok. So, we will not be going into second-order conditions in this course, although you can refer to Nocedal's book, I think chapter 12, which will give you the Hessian-related conditions as well. Alright, so this brings this to an end. The next thing that we will start is we will give you an algorithm to actually solve this problem, which is called the projected gradient descent method.

I already gave an intuitive feel for what projection means, we will formalize that and come up with an algorithm, which will be just as simple for you to code and use in MATLAB as was gradient descent. Ok. So, that will be your algorithm that you can use for solving constraint optimization. Ok, also known as PGD in short. So, everyone knows what the problem is? I have to minimize $x\, f(x)$, where x belongs to some feasible set. As an example of what we will look at, I can have something like this:

$$\min \| Ax - b \|^2$$

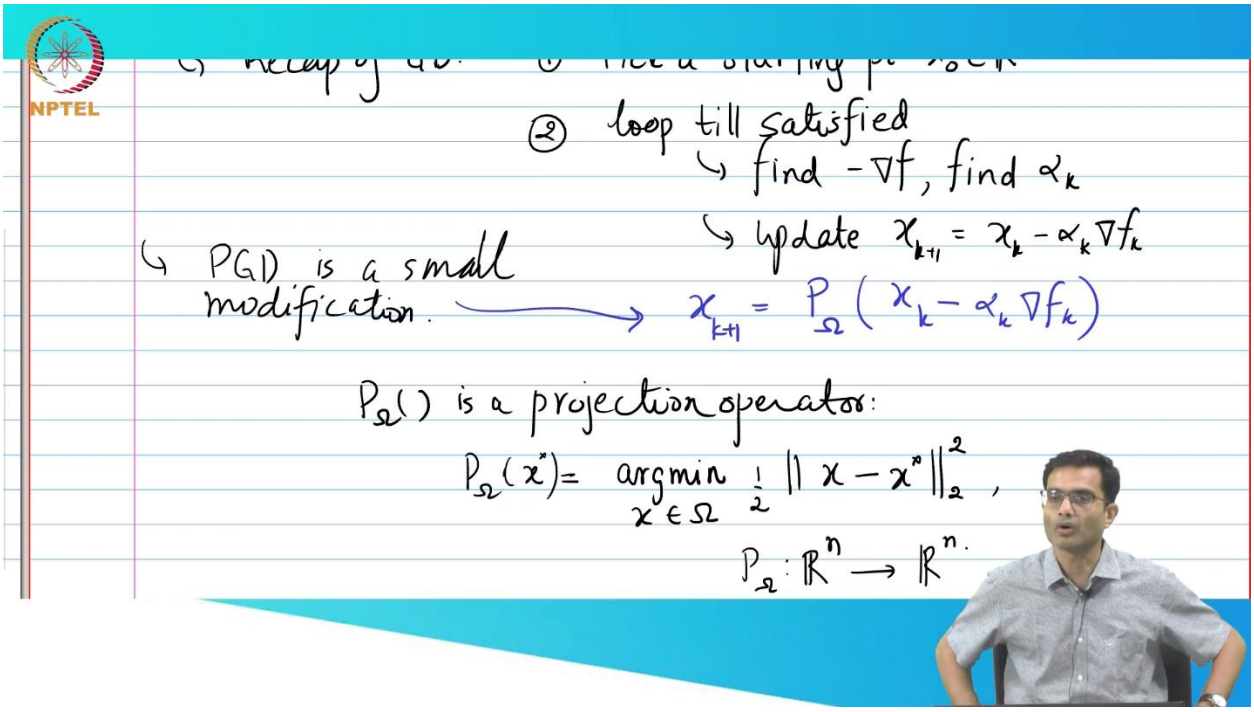
This could be, for example, a linear least squares problem. Right. Now, I can make things a little bit more interesting. I can say something like this:

$$\min \| Ax - b \|^2 \quad \text{such that} \quad \| x \| \leq 1$$

This says, minimize the quantity $Ax - b$ squared such that the length of the solution is inside the unit ball. Right? There, the dimensionality is not specified, so that is why I am calling it an n-dimensional ball. Now, these kinds of problems, you can imagine, if I have constraints on power, my nodes have in a sensor network or an antenna array, they have only maximum unit amplitude that they can put out. This is the kind of problem that I am interested in.

That is what we are going to try to solve. Ok. So, before we get into projected gradient descent, we will quickly review gradient descent and figure out what other small modifications we need to make in order to make it work for a constraint optimization. So, recap of gradient descent:

What are the first things that we did in gradient descent? Pick a random starting point or use some intuition to arrive at a good point. Right? So, let us just say pick, I would not say random, I will just say pick a starting point x_0 that lives in $\mathbb{R}^n$. Ok.



Next, yeah. So, I mean, we need to loop until convergence is satisfied and then within that we will calculate a descent direction and step length. I need to know how much to walk, right? And then I will check if convergence is satisfied or not. Ok. So, loop until satisfied. Ok. So, I am writing it like this. Please do not write this way in the exam. Ok. Loop until satisfied. So, find $-\nabla f$, I need to find that, is the gradient, negative gradient. Find α_k, ok. I have got these and then I update:

$$x_{k+1} = x_k + \alpha_k \cdot (-\nabla f)$$

That is basically it, right? That was all that gradient descent was. Do this, ok. Now, the PGD (projected gradient descent) is a small modification. I do this step, I continue to do gradient descent the way I have done, except once I get x_{k+1}, here is where the problem can happen, right? We can just intuitively think about it. x_k was a point which was feasible, and I am trying to improve it by going to a better point. So, I got my gradient direction, I went in the negative of it, found a step length. What could go wrong? x_{k+1} could jump out of the feasible set. That is the thing that I have to worry about. So, at that point, what I do is I use a projection operation to make sure that I am still feasible. Ok. So, I say:

$$x_{k+1} = P_\omega(x_k - \alpha_k \nabla f_k)$$

This is the, in a nutshell, what projected gradient descent is: you do vanilla gradient descent, but you do a projection step. So, this ensures that x_{k+1} is feasible. Ok, and then that is about it. This surprisingly simple procedure has its own proof of convergence. It works, it actually works. Alright. So, however, there is a catch. What is the catch? This projection operation—what I

showed you a few slides ago where I draw some object and I just draw a perpendicular on it—I can do all of this drawing perpendiculars until class 12, right? I mean, if I have n-dimensional, then what perpendiculars am I drawing, right?

So, this projection operation is actually a mini optimization problem in itself, right? So, let us now write this P operator formally. P is a projection operation operator. So, I give it a point x_0, ok, let us not call it x_0, it is not the starting point, let us call it x^*. Ok. It is an optimization problem. So, I should be able to write it as:

$$\underset{x \in \Omega}{\operatorname{argmin}} \parallel x - x^* \parallel^2$$

What is the objective function that I am trying to minimize? So, you have given me a point x^*, output should be an x^* maybe outside. So, what am I trying to minimize? The distance between x and x^*. Right. So, common sense says that this should be:

$$\min \parallel x - x^* \parallel^2$$

I mean, can I put a square here? Makes no difference, it is the same thing. So, usually, this thing is written as:

$$\frac{1}{2} \parallel x - x^* \parallel^2$$

Just a little bit of convention—the literature likes to put a $\frac{1}{2}$ here. So, this is the projection operation, and this projection operation's input is a point in $\mathbb{R}^n$, output is also a point in $\mathbb{R}^n$. So, the rest of the details of PGD we will discuss subsequently, but this is all that we need to do to get a working constraint optimization algorithm. Goal? Ok, very simple, right? It is simple as long as this subproblem is easy to solve. This may or may not be easy, so we will explore different types of projection operations in what follows.

Ok, any questions on this? Ok, so I will see you guys tomorrow. Well, if you write your projection operation in an efficient way, if you give it a feasible point, without doing much calculation, it will give you the same point back. What is the projection operation applied to a point within Ω? It is the same point. So, if you write it carefully, your algorithm should just send you back that point without wasting too much time.