JAMES SWAN: OK. Well, everyone's quieted down, so that means we have to get started. So let me say something here. This will be our last conversation about optimization. So we've discussed unconstrained optimization. And now we're going to discuss a slightly more complicated problem-- but you're going to see it's really not that much more complicated-- constrained optimization.

These are the things we discussed before. I don't want to spend much time recapping because I want to take a minute and talk about the midterm exam. So we have a quiz. It's next Wednesday. Here's where it's going to be located. Here's 66. Head down Ames. You're looking for Walker Memorial on the third floor.

Unfortunately, the time for the quiz is 7:00 to 9:00 PM. We really did try hard to get the scheduling office to give us something better, but the only way to get a room that would fit everybody in was to do it at this time in Walker. I really don't understand, because I actually requested locations for the quizzes back in April. And somehow I was too early, maybe, and got buried under a pile. Maybe not important enough, I don't know.

But it's got to be from seven to nine next Wednesday. Third floor. There's not going to be any class next Wednesday because you have a quiz instead. So you get a little extra time to relax or study, prepare, calm yourself before you go into the exam. There's no homework this week. So you can just use this time to focus on the material we've discussed.

There's a practice exam from last year posted on the Steller site, which you can utilize and study from. I'll tell you this. That practice exam is skewed a little more towards some chemical engineering problems that motivate the numerics. I've found in the past that when problems like that are given on the exam, sometimes there's a lot of reading that goes into understanding the engineering problem. And that tends to set back the problem-solving.

So I'll tell you that the quiz that you'll take on Wednesday will have less of the engineering associated with it, and focus more on the numerical or computational science. The underlying

sorts of questions, the way the questions are asked, the kinds of responses you're expected to give I'd say are very similar. But we've tried to tune the exam so that it'll be less of a burden to understand the structure of the problem before describing how you'd solve it. So I think that's good.

It's comprehensive up to today. So linear algebra, systems of nonlinear equations and optimization are the quiz topics. We're going to switch on Friday to ordinary differential equations and initial value problems. So you have two lectures on that, but you won't have done any homework. You probably don't know enough or aren't practiced enough to answer any questions intelligently on the quiz. So don't expect that material to be on there. It's not. It's going to be these three topics.

Are there any questions about this that I can answer?

Kristin has a question.

AUDIENCE: [INAUDIBLE].

JAMES SWAN: OK. So yeah, come prepared. It might be cold. It might be hot. It leaks when it rains a little bit. Yeah, it's not the greatest spot. So come prepared. That's true. Other questions? Things you want to know?

AUDIENCE: What can we take to the exam?

JAMES SWAN: Ooh, good question. So you can bring the book recommended for the course. You can bring your notes. You can bring a calculator. You need to bring some pencils. We'll provide blue books for you to write your solutions to the exam in. So those are the materials. Good.

What else? Same question. OK. Other questions? No? OK.

So then let's jump into the topic of the day, which is constrained optimization. So these are problems of the sort. Minimize an objective function f of x subject to the constraint that x belongs to some set D, or find the argument x that minimizes this function. These are equivalent sorts of problem. Sometimes, we want to know one or the other or both. That's not a problem.

And graphically, it looks like this. Here's f, our objective function. It's a nice convex bowl-shaped function here. And we want to know the values of x1 and x2, let's say, that minimize

this function subject to some constraint. That constraint could be that x1 and x2 live inside this little blue circle. It could be D. It could be that x1 and x2 live on the surface of this circle, right, on the circumference of this circle. That could be the constraint.

So these are the sorts of problems we want to solve. D is called the feasible set, and can be described in terms of really two types of constraints. One is what we call equality constraints. So D can be the set of values x such that some nonlinear function c of x is equal to zero. So it's the set of points that satisfy this nonlinear equation. And among those points, we want to know which one produces the minimum in the objective function.

Or it could be an inequality constraint. So D could be the set of points such that some nonlinear function h of x is, by convention, positive. So h of x could represent, for example, the interior of a circle, and c of x could represent the circumference of a circle. And we would have nonlinear equations that reflect those values of x that satisfy those sorts of geometries.

So equality constrained, points that lie on this circle, inequality constrained, points that lie within this circle. The shape of the feasible set is constrained by the problem that you're actually interested in. So it's easy for me to draw circles in the plane because that's a shape you're familiar with. But actually, it'll come from some sort of physical constraint on the engineering problem you're looking at, like mole fractions need to be bigger than zero and smaller than one, and temperatures in absolute value have to be bigger than zero and smaller than some value because that's a safety factor on the process.

So these set up the constraints on various sorts of optimization problems that we're interested in. It could also be true that we're interested in, say, optimization in the domain outside of this circle, too. It could be on the inside, could be on the outside. That's also an inequality constrained sort of problem. You know some of these already. They're familiar to you.

So here's a classic one from mechanics. Here's the total energy in a system for, say, a pendulum. So x is like the position of the tip of this pendulum and v is the velocity that it moves with. This is the kinetic energy. This is the potential energy. And we know the pendulum will come to rest in a place where the energy is minimized.

Well, the energy can only be minimized when the velocity here is zero, because any non-zero velocity will always push the energy content up. So it comes to rest. It doesn't move. And then there's some value of x at which the energy is minimized. If there is no constraint that says that the pendulum is attached to some central axis, then I can always make the energy smaller

by making x more and more negative. It just keeps falling. There is no stopping point.

But there's a constraint. The distance between the tip of the pendulum and this central point is some fixed distance out. So this is an equality constrained sort of problem, and we have to choose from the set of v and x the values subject to this constraint that minimize the total energy. And that's this configuration of the pendulum here. So you know these sorts of problems already.

We talked about this one, linear sorts of programs. These are optimization problems where the objective function is linear in the design variables. So it's just the dot product between x and some vector c that weights the different design options against each other. So we talked about ice cream. Yes, this is all premium ice cream because it comes in the small containers, subject to different constraints.

So those constraints can be things like, oh, x has to be positive because we can't make negative amounts of ice cream. And maybe we've done market research that tells us that the market can only tolerate certain ratios of different types of ice cream. And that may be some set of linear equations that describe that market research that sort of bound the upper values of how much ice cream we can put out on the market. And then we try to choose the optimal blend of pina colada and strawberry to sell.

So those are linear programs. This is an inequality constrained optimization. In general, we might write these problems like this. We might say minimize f of x subject to the constraint that c of x is 0 and h of x is positive. So minimize it over the values of x that satisfy these two constraints.

There's an old approach that's discussed in the literature. And it's not used. I'm going to describe it to you, and then I want you to try to figure out why it's not used. And it's called the penalty method. And the penalty method works this way. It says define a new objective function, which is our old objective function plus some penalty for violating the constraints.

How does that penalty work? So we know that we want values of x for which c of x is equal to 0. So if we add to our objective function the norm of c of x-- this is a positive quantity-- this is a positive quantity-- whenever x doesn't satisfy the constraint, this positive quantity will give us a bigger value for this objective function f than if c of x was equal to 0.

So we penalize points which don't satisfy the constraint. And in the limit that this penalty factor

mu here goes to zero, the penalties get large, so large that our solution will have to prefer satisfying the constraints. There's another penalty factor over here, which is identical to this one but for the inequality constraint. It says take a heaviside step function for which is equal to 1 when the value of its argument is positive, and it's equal to zero when the value of its argument is negative.

So whenever I violate each of my inequality constraints, Hi of x, turn on this heaviside step function, make it equal to 1, and then multiply it by the value of the constraint squared, a positive number. So this is the inequality constraint penalty, and this is the equality constraint penalty. People don't use this, though.

It makes sense. I take the limit that mu goes to zero. I'm going to have to prefer solutions that satisfy these constraints. Otherwise, if I don't satisfy these constraints, I could always move closer to a solution that satisfies the constraint, and I'll bring down the value of the objective function. I'll make it lower. So I'll always prefer these lower value solutions.

But can you guys take a second and sort of talk to each other? See if you can figure out why one doesn't use this method. Why is this method a problem?

OK, I heard the volume go up at some point, which means either you switched topics and felt more comfortable talking about that than this, or maybe you guys were coming to some conclusions, or had some ideas about why this might be a bad idea. Do you want to volunteer some of what you were talking about? Yeah, Hersh.

AUDIENCE: Could it be that [INAUDIBLE]?

JAMES SWAN: Well, that's an interesting idea. So yeah, if we have a non-convex optimization problem, there could be some issues with f of x, and maybe f of x runs away so fast that I can never make the penalty big enough to enforce the constraint. That's actually a really interesting idea. And I like the idea of comparing the magnitude of these two terms. I think that's on the right track.

Were there some other ideas about why you might not do this? Different ideas? Yeah.

AUDIENCE: [INAUDIBLE].

JAMES SWAN: Well, you know, that that's an interesting idea, but actually the two terms in the parentheses here are both positive. So they're only going to be minimized when I satisfy the constraints. So the local minima of the terms in parentheses sit on or within the boundaries of the feasible set

that we're looking at. So by construction, actually, we're going to be able to satisfy them because the local minima of these points sits on these boundaries. These terms are minimized by satisfying the constraints. Other ideas? Yeah.

AUDIENCE: Do your iterates have to be feasible?

JAMES SWAN: What's that?

AUDIENCE: Your iterates don't have to be feasible?

JAMES SWAN: Ooh, this is a good point. The iterates-- this is an unconstrained optimization problem. I'm just going to minimize this objective function. It's like what Hersh said, I can go anywhere I want in the domain. I'm going to minimize this objective function, and then I'm going to try to take the limit as mu goes to zero. The iterates don't have to be feasible. Maybe I can't even evaluate f of x if the iterates aren't feasible. That's an excellent point. That could be an issue.

Anything else? Are there some other ideas? Sure.

AUDIENCE: [INAUDIBLE].

JAMES SWAN: I think that's a good point.

AUDIENCE: --boundary from outside without knowing what's inside.

JAMES SWAN: Short. So you'll see, actually, the right way to do this is to use what's called interior point methods, which live inside the domain. This is an excellent point. There's another issue with this that's I think actually less subtle than some of these ideas, which they're all correct, actually. These can be problems with this sort of penalty method.

As I take the limit that mu goes to zero, the penalty function becomes large for all points outside the domain. They can become larger than f for those points. And so there are some practical issues about comparing these two terms against each other. I may not have sufficient accuracy, sufficient number of digits to accurately add these two terms together. So I may prefer to find some point that lives on the boundary of the domain as mu goes to zero. But I can't guarantee that it was a minima of f on that domain, or within that feasible set.

So a lot of practical issues that suggest this is a bad idea. This is an old idea. People knew this was bad for a long time. It seems natural, though. It seems like a good way to transform from these constrained optimization problems to something we know how to solve, an

unconstrained optimization. But actually, it turns out not to be such a great way to do it.

So let's talk about separating out these two different methods from each other, or these two different problems. Let's talk first about equality constraints, and then we'll talk about inequality constraints. So equality constrained optimization problems look like this. Minimize f of x subject to c of x equals zero.

And let's make it even easier. Rather than having some vector of equality constraints, let's just have a single equation that we have to satisfy for that equality constraint, like the equation for a circle. Solutions have to sit on the circumference of a circle. So one equation that we have to satisfy.

You might ask again, what are the necessary conditions for defining a minimum? That's what we used when we had equality-- or when we had unconstrained optimization. First we had to define what a minimum was, and we found that minima were critical points, places where the gradient of the objective function was zero. That doesn't have to be true anymore. Now, the minima has to live on this boundary of some domain. It has to live in this set of points c of x equals zero. And the gradient of f is not necessarily zero at that minimal point.

But you might guess that Taylor expansions are the way to figure out what the appropriate conditions for a minima are. So let's take f of x, and let's expand it, do a Taylor expansion in some direction, d. So we'll take a step away from x, which is small, in some direction, d. So f of x plus d is f of x plus g dot d, the dot product between the gradient of f and d.

And at a minimum, either the gradient is zero or the gradient is perpendicular to this direction we moved in, d. We know that because this term is going to increase-- well, will change the value of f of x. It will either make it bigger or smaller depending on whether it's positive or negative. In either case, it will say that this point x can't be a minimum unless this term is exactly equal to zero in the limit that d becomes small. So either the gradient is zero or the gradient is orthogonal to this direction d we stepped in. And d was arbitrary. We just said take a step in a direction, d.

Lets take our equality constraint and do the same sort of Taylor expansion, because we know if we're searching for a minima along this curve c of x better be equal to zero. It better satisfy the constraint. And also, c of x plus d, that little step in the direction d, should also satisfy the constraint. We want to study only the feasible set of values.

So actually, d wasn't arbitrary. d had to satisfy this constraint that, when I took this little step, c of x plus d had to be equal to zero. So again, we'll take now a Taylor expansion of c of x plus d, which is c of x plus grad of c of x dotted with d. And that implies that d must be perpendicular to the gradient of c of x, because c of x plus d has to be zero and c of x has to be zero. So the gradient of c of x dot d-- it's a leading order has also got to be equal to zero.

So d and the gradient in c are perpendicular, and d and the gradient in g have to be perpendicular at a minimum. That's going to define the minimum on this equality constrained set. Does that make sense? c satisfies the constraint, c plus d satisfies the constraint. If this is true, d has to be perpendicular to the gradient of c, g has to be perpendicular to the gradient of d. d is, in some sense, arbitrary still. d has to satisfy condition that it's perpendicular to the gradient of c, but who knows, there could be lots of vectors that are perpendicular to the gradient of c.

So the only generic relationship between these two we can formulate is g must be parallel to the gradient of c. g is perpendicular to d, gradient of c is perpendicular to d. In the most generic way, g and gradient of c should be parallel to each other, because d I can select arbitrarily from all the vectors of the same dimension as x.

If g is parallel to the gradient of c, then I can write that g minus some scalar multiplied by the gradient of c is equal to zero. That's an equivalent statement, that g is parallel to the gradient of c. So that's a condition associated with points x that solve this equality constrained problem. The other condition is that point x still has to satisfy the equality constraint.

But I introduced a new unknown, this lambda, which is called the Lagrange multiplier. So now I have one extra unknown, but I have one extra equation. Let me give you a graphical depiction of this, and then I'll write down the formal equations again.

So let's suppose we want to minimize this parabolic function subject to the constraint that the solution lives on the line. So here's the contours of the function, and the solution has to live on this line. So I get to stand on this line, and I get to walk and walk and walk until I can't walk downhill anymore. and I've got to turn and walk uphill again.

And you can see the point where I can't walk downhill anymore is the place where this constraint is parallel to the contour, or where the gradient of the objective function is parallel to the gradient of the constraint. So you can actually find this point by imagining yourself moving along this landscape. After I get to this point, I start going uphill again.

So that's the method of Lagrange multipliers. Minimize f of x subject to this constraint. The solution is given by the point x at which the gradient is parallel to the gradient of c, and at which c is equal to zero. And you solve this system of nonlinear equations for two unknowns. One is x, and the other is this unknown lambda. How far stretched is the gradient in f relative to the gradient in c?

So again, we've turned the minimization problem into a system of nonlinear equations. In order to satisfy the equality constraint, we've had to introduce another unknown, the Lagrange multiplier. It turns out this solution set, x and lambda, is a critical point of something called the Lagrangian. It's a function f of x minus lambda times c.

It's a critical point in x and lambda of this nonlinear function called the Lagrangian. It's not a minimum of this function, unfortunately. It's a saddle point of the Lagrangian, it turns out. So we're trying to find a saddle point of the Lagrangian. Does this make sense? Yes? OK.

We've got to be careful, of course. Just like with unconstrained optimization, we actually have to check that our solution is a minimum. We can't take for granted, we can't suppose that our nonlinear solver found a minimum when it solved this equation. Other critical points can satisfy this equation, too. So we've got to go back and try to check robustly whether it's actually a minimum.

But this is the method. Introduce an additional unknown, the Lagrange multiplier, because you can show geometrically that the gradient of the objective function should be parallel to the gradient of the constraint at the minimum. Does that make sense? Does this picture make sense? OK. So you know how to solve systems of nonlinear equations, you know how to solve constrained optimization problems.

So here's f. Here's c. We can actually write out what these equations are. So you can show that the gradient of x minus lambda gradient of c, that's a vector, 2x1 minus lambda and 20x2 plus lambda. And c is the equation for this line down here, so x1 minus x2 minus 3. And that's all got to be equal to zero. In this case, this is just a system of linear equations. So you can actually solve directly for x1, x2, and lambda. And it's not too difficult to find the solution for all three of these things by hand.

But in general, these constraints can be nonlinear. The objective function doesn't have to be quadratic. Those are the easiest cases to look at. And the same methodology applies. And so

you should check that you're able to do this. This is the simplest possible equality constraint problem. You could do it by hand. You should check that you're actually able to do it, that you understand the steps that go into writing out these equations.

Let's just take one step forward and look at a less generic case, one in which we have a vector valued function that gives the equality constraints instead. So rather than one equation we have to satisfy, there may be many. It's possible that the feasible set doesn't have any solutions in it. It's possible that there is no x that satisfies all of these constraints simultaneously. That's a bad problem to have. You wouldn't like to have that problem very much. But it's possible that that's the case.

But let's assume that there are solutions for the time being. So there are x's that satisfy the equality constraint. Let's see if we can figure out again what the necessary conditions for defining a minima are. So same as before, let's Taylor expand f of x going in some direction, d. And let's make d a nice small step so we can just treat the f of x plus d as a linearized function.

So we can see again that g has to be perpendicular to this direction, d, if we're going to have a minima. Otherwise, I could step in some direction, d, and I'll find either a smaller value of f of x plus d or a bigger value of f of x plus d. So g has to be perpendicular to d. And for the equality constraints, again, they all have to satisfy this equality constraint up there. So c of x has to be equal to zero, and c of x plus d also has to be equal to zero.

And so if we take a Taylor expansion of c of x plus d, about x, you'll get c of x plus d plus the Jacobian of c, all the partial derivatives of c with respect to x, multiplied by d. We know that c of x plus d is zero, and c of x is zero, so the directions, d, belong to what set of vectors? The null space.

So these directions have to live in the null space of the Jacobian of c. So I can't step in any direction, I have to step in directions that are in the null space of c. g is perpendicular to d, as well. And d belongs to the null space of c. In fact, you know that d is perpendicular to each of the rows of the Jacobian. Right? You know that? I just do the matrix vector product, right?

And so each element of this matrix vector product is the dot product of d with a different row of the Jacobian. So those rows are a set of vectors. Those rows describe the range of J transpose, or the row space of J. Remember we talked about the four fundamental subspaces, and I said we almost never use those other ones, but this is one time when we will.

So those rows belong to the range of J transpose, or they belong to the left null space of J. I need to find a g, a gradient, which is always perpendicular to d. And I know d is always perpendicular to the rows of J. So I can write g as a linear superposition of the rows of J. As long as g is a linear superposition of the rows, it'll always be perpendicular to d. Vectors from the null space of a matrix are orthogonal to vectors from the row space of that matrix, it turns out. And they're orthogonal for this reason.

So it tells us, if Jd is zero, then d belongs to the null space. g is perpendicular to d. That means I could write g as a linear superposition of the rows of J. So g belongs to the range of J transpose, or it belongs to the row space of J. Those are equivalent statements. And therefore, I should be able to write g as a linear superposition of the rows of J.

And one way to say that is I should be able to write g as J transpose times some other vector lambda. That's an equivalent way of saying that g is a linear superposition of the rows of J. I don't know the values of lambda. So I introduced a new set of unknowns, a set of Lagrange multipliers. My minima is going to be found when I satisfy this equation, just like before, and when I'm able to satisfy all of the equality constraints.

How many Lagrange multipliers do I have here? Can you figure that out? You can talk with your neighbors if you want. Take a couple minutes. Tell me how many Lagrange multipliers, how many elements are in this vector lambda.

How many elements are in lambda? Can you tell me? Sam.

AUDIENCE: Same as the number of equality constraints.

JAMES SWAN: Yes. It's the same as the number of equality constraints. J came from the gradient of c. It's the Jacobian of c. So it has a number of columns equal to the number of elements in x, because I'm taking partial derivatives with respect to each element of x, and has a number of rows equal to the number of elements of c.

So J transpose, I just transpose those dimensions. And lambda must have the same number of elements as c does in order to make this product make sense. So I introduce a new number of unknowns. It's equal to exactly the number of equality constraints that I had, which is good, because I'm going to make a system of equations that says g of x minus J transpose lambda equals 0 and c of x equals 0.

And the number of equations here is the number of elements in x for this gradient, and the number of elements in c for c. And the number of unknowns is the number of elements in x, and the number of elements in c associated with the Lagrange multiplier. So I have enough equations and unknowns to determine all of these things.

So whether I have one equality constraint or a million equality constraints, the problem is identical. We use the method of Lagrange multipliers. We have to solve an augmented system of equations for x and this projection on the row space of J, which tells us how the gradient is stretched or made up, composed of elements of the row space of J. These are the conditions associated with a minima in our objective function on this boundary dictated by the equality constraint.

And of course, the solution set is a critical point of a Lagrangian, which is f of x minus c dot lambda. And it's not a minimum of it, it's a critical point. It's a saddle point, it turns out, of this Lagrangian. So we've got to check, did we find a saddle point or not when we find a solution to this equation here. But it's just a system of nonlinear equations.

If we have some good initial guess, what do we apply? Newton-Raphson, converge rate towards the solution. If we don't have a good initial guess, we've discussed lots of methods we could employ, like homotopy or continuation to try to develop good initial guesses for what the solution should be. Are there any questions about this? Good.

OK. So you go to Matlab and you call fmincon, do a minimization problem, and you give it some constraints. Linear constraints, nonlinear constraints, it doesn't matter actually. The problem is the same for both of them. It's just a little bit easier if I have linear constraints. If this constraining function is a linear function, then the Jacobian I know. It's the coefficient matrix of this linear problem. Now I only have to solve linear equations down here.

So the problem is a little bit simpler to solve. So Matlab sort of breaks these apart so it can use different techniques depending on which sort of problem is posed. But the solution method is the same. It does the method of Lagrange multipliers to find the solution. OK?

Inequality constraints. So interior point methods were mentioned. And it turns out this is really the best way to go about solving generic inequality constrained problems. So the problems of the sort minimize f of x subject to h of x is positive, or at least not negative. This is some nonlinear inequality that describes some domain and its boundary in which the solution has to live.

And what's done is to rewrite as an unconstrained optimization problem with a barrier that's incorporated. This looks a lot like the penalty method, but it's very different. And I'll explain how. So instead, we want to minimize this f of x minus mu times the sum of log of h, each of these constraints. If h is negative, we'll take the log of the negative argument. That's a problem computationally.

So the best we could do is approach the boundary where h is equal to zero. And as h goes to zero, the log goes to minus infinity. So this term tends to blow up because I've got a minus sign in front of it. So this is sort of like a penalty, but it's a little different because the factor in front I'm actually going to take the limit as mu goes to zero. I'm going to take the limit as this factor gets small, rather than gets big.

The log will always get big as I approach the boundary of the domain. It'll blow up. So that's not a problem. But I can take the limit that mu gets smaller and smaller. And this quantity here will have less and less of an impact on the shape of this new objective function and mu gets smaller and smaller. The impact will only be nearest the boundary. Does that make sense?

So you take the limit that mu approaches zero. It's got to approach it from the positive side, not the negative side, so everything behaves well. And this is called an interior point method. So we have to determine the minimum of this new objective function for progressively weaker barriers. So we might start with some value of mu, and we might reduce mu progressively until we get mu down small enough that we think we've converged to a solution.

So how do you do that reliably? What's the procedure one uses to solve a problem successively for different parameter values?

AUDIENCE: [INAUDIBLE].

JAMES SWAN: Yeah, it's a homotopy, right? You're just going to change the value of this barrier parameter. And you're going to find a minima. And if you make a small change in the barrier parameter, that's going to serve as an excellent initial guess for the next value. And so you're just going to take these small steps. And the optimization routine is going to carry you towards the minimum in the limit that mu goes to zero. So you do this with homotopy.

Here's an example of this sort of interior point method, a trivial example. Minimize x subject to x being positive. So we know the solution lives where x equals zero. But let's write this as

unconstrained optimization using a barrier. So minimize x minus mu times log x. Here's x minus mu times log x. So out here, where x is big, x wins over log x, so everything starts to look linear. But as x become smaller and smaller, log x gets very negative, so minus log x gets very positive. And here's the log creeping back up.

And as I decrease mu smaller and smaller, you can see the minima of this function is moving closer and closer and closer to zero. So if I take the limit that mu decreases from some positive number towards zero, eventually this minimum is going to converge to the minimum of the constrained inequality, constrained optimization problem. Make sense? OK.

OK. So we want to do this. You can use any barrier function you want. Any thoughts on why a logarithmic barrier is used?

No. OK, that's OK. So minus log is going to be convex. Log isn't convex, but minus log is going to be convex. So that's good. If this function's convex, then their combination's going to be convex, and we'll be OK. But the gradient of the log is easy to compute. Grad log h is 1 over h grad h. So if I know h, I know grad h, it's easy for me to compute the gradient of log h.

We know we're going to solve this unconstrained optimization problem where we need to take grad of this objective function equal zero. So the calculations are easy. The log makes it easy like that. The log is also like the most weakly singular function available to us. Out of all the tool box of all problems we can reach to, the log has the mildest sort of singularities. Singularities at both ends, which is sort of funny, but the mildest sort of singularities you have to cope with.

So we want to find the minimum of these unconstrained optimization problems where the gradient of f minus mu sum 1 over h, grad h, is equal to zero. And we just do that for progressively smaller values of mu, and we'll converge to a solution. That's the interior point method.

You use homotopy to study a sequence of barrier parameters, or continuation to study a sequence of barrier parameters. You stop the homotopy or continuation when what? How are you going to stop?

I've got to make mu small, right? I want to go towards the limit mu equals zero. I can't actually get to mu equals zero, I've just got to approach it. So how small do I need to make mu before I quit? It's an interesting question. What do you think?

I'll take this answer first.

AUDIENCE: So it doesn't affect the limitation.

JAMES SWAN: Good. So we might look at the solution and see is the solution becoming less and less sensitive to the choice of mu. Did you have another suggestion?

AUDIENCE: [INAUDIBLE].

JAMES SWAN: Set the tolerance. Right, OK.

AUDIENCE: [INAUDIBLE].

JAMES SWAN: Mhm. Right, right, right, right. So you--

AUDIENCE: [INAUDIBLE].

JAMES SWAN: Good. So there were two suggestions here. One is along the lines of a step-norm criteria, like I check my solution as I change mu, and I ask when does my solution seem relatively insensitive to mu. When the changes in these steps relative to mu get sufficiently small, I might be willing to accept these solutions as reasonable solutions for the constrained optimization.

I can also go back and I can check sort of function norm criteria. I can take the value of x I found as the minimum, and I can ask how good a job does it do satisfying the original equations. How far away am I from satisfying the inequality constraint? How close am I to actually minimizing the function within that domain?

OK. So we're running out of time here. Let me provide you with an example.

So let's minimize again-- I always pick this function because it's easy to visualize, a nice parabolic function that opens upwards. And let's minimize it subject to the constraint that h of x1 and x2 is equal to 1 minus-- well, the equation for a circle of radius 1, essentially. The interior of that circle. So here's the contours of the function, and this red domain is the constraint. And we want to know the smallest value of f that lives in this domain.

So here's a Matlab code. You can try it out. And make a function, the objective function, f, it's x squared plus 10x-- x1 squared plus 10x2 squared. Here's the gradient. Here's the Hessian. Here, I calculate h. Here's the gradient in h. Here's the Hessian in h.

I've got to define a new objective function, phi, which is f minus mu log h. This is the gradient in phi and this is the Hessian of phi. Oh, man, what a mess. But actually, not such a mess, because the log makes it really easy to take these derivatives. So it's just a lot of differential sort of calculus involved in working this out, but this is the Hessian of phi.

And then I need some initial guess. So I pick the center of my circle as an initial guess for the solution. And I'm going to loop over values of mu that get progressively smaller. I'll just go down to 10 to the minus 2 and stop for illustration purposes here. But really, we should be checking the solution as we go and deciding what values we want to stop with.

And then this loop here, what's this do? What's it do? Can you tell?

AUDIENCE: Is it Newton?

JAMES SWAN: What's that?

AUDIENCE: Newton?

JAMES SWAN: Yeah, it's Newton-Raphson, right? x is x minus Hessian inverse times grad phi, right? So I just do Newton-Raphson. I take my initial guess and I loop around with Newton-Raphson, and when this loop finishes, I reduce mu, and it'll just use my previous guess as the initial guess for the next value of the loop, until mu is sufficiently small. OK? Interior point method.

Here's what that solution path looks like. So mu started at 1, and the barrier was here. It was close to the edge of the circle, but not quite on it. But as I reduced mu further and further and further, you can see the path, the solution path, that was followed works its way closer to the boundary of the circle. And the minimum is found right here. So it turns out the minimum of this function doesn't live in the domain, it lives on the boundary of the domain.

Recall that this point should be a point where the boundary of the domain is parallel to the contours of the function, since actually we didn't need the inequality constraint here. We could have used the equality constraint. The equality constrained problem has the same solution as the inequality constrained problem. And look, that actually happened. Here's the contours of the function. The contour of the function runs right along here, and you can see it looks like it's going to be tangent to the circle at this point. So the interpoint method actually solved an equality constrained problem in addition to an inequality constrained problem, which is-- that's sort of cool that you can do it that way.

How about if I want to do a combination of equality and inequality constraints? Then what do I do? Yeah.

AUDIENCE: [INAUDIBLE].

JAMES SWAN: Perfect. Convert the equality constraint into unknowns, Lagrange multipliers, instead. And then do the interior point method on the Lagrange multiplier problem. Now you've got a combination of equality and inequality constrained. This is exactly what Matlab does. So it converts equality constraints into Lagrange multipliers. Inequality constraints it actually solves using interior point methods. Buried in that interior point method is some form of Newton-Raphson and steepest descent combined together, like dog leg we talked about for unconstrained problems.

And it's going to do a continuation. As it reduces the values of mu, it'll have some heuristic for how it does that. It's going to use its previous solutions as initial guesses for the next iteration. So these are very complicated problems, but if you understand how to solve systems of nonlinear equations, and you think carefully about how to control numerical error in your algorithm, you come to a conclusion like this, that you can do these sorts of Lagrange multiplier interior point methods to solve a wide variety of problems with reasonable reliability. OK? Any more questions?

No? Good. Well, thank you, and we'll see you on Friday.