

The following content is provided under a Creative Commons license. Your support will help MIT OpenCourseWare continue to offer high quality educational resources for free. To make a donation or to view additional materials from hundreds of MIT courses, visit MIT OpenCourseWare at ocw.mit.edu.

PROFESSOR:

OK, let's go ahead and get started again. So we had a nice review on Monday of some of the different concepts that we've talked about up to this point in the course. We've covered linear algebra, solutions of systems of nonlinear equations. There's been a lot of fundamentals involved in that.

Things are going to get more applied as the sorts of problems we're trying to solve get more complicated. Complicated problems require very sophisticated methods to solve them. It's not going to be our expectation that you're necessarily able to implement all of these sophisticated methods.

So for example, you implemented Newton's method on your last homework assignment. That'll be the last time we ask you to do that by hand. You'll utilize tools built into Matlab to solve other systems of nonlinear equations going forward.

You understand how it works and what the limitations are. You understand how to build different tools that will be inputs to these functions like the function you're trying to find the root for, or an explicit expression for the Jacobian.

You even understand how to set certain parameters associated with these methods, and what those particular parameter values mean. Like setting the function norm tolerance, or setting the step norm tolerance associated with these methods.

Setting the number of iterations you're going to let the method go. You've implemented it, so you understand what these things mean, what the consequences of choosing them are, how they might increase the time associated with computations that you're going to conduct.

So we've done a lot of fundamentals. Maybe that was difficult for some of you who haven't seen this before. Now things are becoming more about the practice of numerical methods. How do we actually solve things reliably, and that's what today's lecture will talk about.

We discussed solving systems of nonlinear equations and needing good initial guesses for our

methods. Bad initial guesses can lead to unpredictable results. Good initial guesses, we know the Newton-Raphson method is going to double the number of accurate digits each iteration, and we should converge very rapidly to a solution.

So it seems like having good initial guesses is really ideal. And this topic, homotopy and bifurcation, will cover precisely those issues, and the practice of getting these good initial guesses. I think is one of the most useful things you'll learn in this class, actually. It's a really nice way of going about solving complicated problems.

Before we begin, I want to address a question I got via email last night. That's always good if these questions go on [? Piazza ?] so everyone can read them, but I thought it was a very nice question about how do you find-- the question was how do you find multiple roots using the Newton-Raphson method.

So if I have a polynomial, where the function doesn't cross through the origin, but it just touches the origin, and goes back up. That's a root with multiplicity in the polynomial. And can the Newton-Raphson method find that? Because the derivative of the function is 0 at the root.

So who says the Newton-Raphson method will be able to find those roots? What do you think? Yes? And who says it won't be able to find those roots? Or who doesn't know? You don't know. That's OK. That's a good question, right. Do you know or don't you know?

So we know the Newton-Raphson method relies on knowing the derivative of the function at different points in order to take our next Newton-Raphson step towards that solution. It may be the case, the derivative is 0 at the root, but nearby the root, the derivative may not be 0.

And then the Newton-Raphson steps are going to be fine. They're always going to keep stepping down towards the root. If you go through an analysis of how the algorithm converges, you'll find out that you only get linear convergence in that case, instead of quadratic convergence.

So there will be some penalty to pay for the fact that the derivative is 0 at the root itself, but it's not going to be so bad. The method can still converge to those sorts of solutions. Not all is lost there, but the rate of convergence may be not as good as you expected it to be.

And the same is true in the multi-dimensional case as well. It's a little hard to imagine what a multiple root is going to look like. It's something like tangent curves in the solution of plane. So

if I have f_1 of x_1 and x_2 equal 0, that's a curve in the x_1, x_2 plane. And I have f_2 in that same plane equal to 0. The place where they touch might be one of these multiple roots. That's a place where the determinant of the Jacobian may be equal to 0.

You'll have the same sort of circumstance. Nearby, the Jacobian can be non singular, and you can take steps towards the root, but there's a worst case scenario where you're coming in on a direction that's parallel that belongs to the null space of the Jacobian at the root, and then everything's going to slow down. You can still converge, but it may be very slow as a consequence.

I thought that was a really nice question. It's a detailed question about how Newton-Raphson works. It's not necessarily going to fail under these weird circumstances. They don't come up too often. They're going to come up in our discussion today, actually. It's not going to fail, but it may converge very slowly to the solution. OK? Good.

I'm going to remind you, the last thing we did with Newton-Raphson methods, we talked about one way of fixing them. In lieu of having good initial guesses, we may not want to take the full Newton-Raphson step. We may want to do these quasi Newton-Raphson methods instead. One of those was this damp Newton-Raphson where we introduced some damping parameter.

We don't take the full step, we take some shorter step, and we try to make the absolute value over function become as small as possible. That's an optimization problem. It's as hard as finding the root of the system of equations that we're looking at. So we don't solve that optimization problem exactly. That's crazy.

Instead we use this backtracking line search method where we change our damping parameter in some systematic way until we get to a point where we're satisfied. Where we say OK, in the process of changing this, we've actually found a smaller value of our function, and we accept that as our best guess for where the next iterate is supposed to go.

And this sort of approach can give you global convergence for the algorithm, which is great, but it's not globally convergent to roots. It's globally convergent to minima, asymptotes, and roots as well. But it's globally convergent. So it'll terminate. The algorithm will stop, and when it stops, you can ask is this a root or not? Do I think this is sufficiently close to a root or not?

If it's not sufficiently close to a root, then you know you terminated somewhere like a minima,

or you rushed out towards an asymptote, and maybe you need to start with a new initial guess for your solution. These quasi Newton-Raphson methods are very useful. This is what Matlab uses in order to-- it uses a more sophisticated version of this called the Dogleg method, but it's the same principle.

It's trying to make sure you're not taking steps that are too big. Trying to make sure you're taking steps in the right direction so that you're approaching the roots of your function. Or at least reducing the absolute value of your function as close to zero as you can get it. Are there any questions about this? Anything we need to follow up on in the previous lecture? No.

OK, good initial guesses. This is the key, really, for solving systems of nonlinear equations. It's the key for doing optimization problems too. We need to have some idea of what our solution looks like, and we want guesses that are close to those solutions so that maybe we are in this region of local convergence for the Newton-Raphson method, and everything converges very quickly.

So where do they come from? This is an important question, and I'll show you how you can get them. There's another issue too, which is related to this, which is the non-linear equations, they can have multiple roots. If we do optimization problems-- that's our next topic-- they can have multiple minima or maxima that we're trying to seek out.

Depending on our initial guess, our numerical method may find one of these roots or another root. It may find one of these minima or another minima. Maybe we're looking for the global minimum, so we need to look at them all. Are there methods that we can use to find them all? So we'll talk about continuation in homotopy, and we'll talk about bifurcation.

And it turns out there are methods one can use to find all the roots for a system of non-linear equations. In fact, the homework assignment you have this week will ask you to do ternary phase equilibrium problem. Vapor liquid phase equilibrium with three components.

There are five parts to this problem. Four of them, I think, are relatively straightforward. They're about finding the pure component roots or randomly guessing some initial conditions for your nonlinear solver in order to find all of the different azeotropes in the system.

There's a fifth part that asks you to actually implement this homotopy method, and try to find all the roots in one go with your algorithm. Like the other homeworks you've had, that part's a little more tricky. Usually these homeworks have one thing that's a little bit more tricky than

everything else.

That part is a little bit more tricky, but it's really instructive to try to do it. We'll learn how to do it today. You can find all of the coexisting compositions of this ternary phase diagram, simultaneously, with one go through this algorithm.

So let's talk about continuation first. This is an interesting idea, and it relates to trying to find roots of these equations, having good initial guesses. Here's a simple example. We have a third degree polynomial, and it has three roots. We can look, graphically, because this is one dimension, and see that one of these roots is near minus 1 and 1/2, another root is near a 1/2, and another root is near 1.

And so I can go to my non-linear equation solver, and give it initial guesses, minus 1/2, 1/2, and 1, and I'm probably going to find these roots pretty reliably. But in many dimensions, we can't do these sorts of graphical approaches. You can't look at this function. It may be difficult to actually get enough data to even look at it if it's a two-dimensional set of functions.

It's hard to see where these roots are, and so continuation is a method of transforming the problem into an easier one to solve. So let's take this problem, this third order polynomial, and let's transform it. Let's replace the 2 with a 2 times the lambda, and let's try to find the roots of this polynomials as lambda grows from 0 to 1.

This is sort of a weird idea, but we actually know when lambda is equal to 0, this goes away, and there's only one root to this polynomial-- one real root-- which is minus 1. We actually know the answer when lambda is equal to 0.

If I make lambda a little bit bigger than 0, there will be a root which is close to minus 1. In fact, minus 1 is a good initial guess for that root, and my non-linear equation solver ought to converge very quickly to that value. If I choose lambda to be small, then it should be in that region of local convergence of my Newton-Raphson method, and I very quickly should find the next root. So I can step lambda along.

I use the lambda equals 0 root as an initial guess for the next root, and I keep increasing lambda from 0 all the way up to 1, and when lambda is equal to 1, I'm going to find the root to the initial problem I was looking at. Does that makes sense? So I'm continuing from one set of solutions to the next. I vary this parameter continuously, and for previous values of that parameter, they serve as initial guesses for the next solution.

Here's what a code that does that would look like. I define my lambdas. We don't know good ways to define the lambda. I've got to figure that out for myself, but here lambda is a vector that goes from 0 to 1, in 1/100 increments. Maybe that's too fine, but that's OK.

And my initial guess for the solution when lambda equals 0 will be minus 1, and I loop over all the values of lambda, and I use `f0`, which is the one-dimensional non-linear equation solver in Matlab, to find the roots of this polynomial function-- $x^3 - 2\lambda x + 1$. Here's the solution and that becomes my initial guess for my next solution.

So this loop is going to go round and round until I get lambda equals 1, and the solution that results there should be the solution to the initial problem I was interested in. Here's what that looks like graphically. I started with lambda equals 0. The root there serves as an initial guess for lambda equals 1/100, and so on, and these roots converge eventually, to something that's close to minus 1 and 1/2.

So the actual value is a little bit bigger or a little bit smaller than minus 1.6. Here's one of the roots. So rather than having to have a precise initial guess, I transformed my problem from an easy to solve one to a more difficult to solve one, and I used the path along that transformation to give me rapid convergence to better and better initial guesses to the problem that I was interested in. Make sense?

There's another way of doing this too. You don't have to go from 0 to 1. Let's go from lambda equals something big back down to 1 instead. So if lambda's big, this term dominates the polynomial equation. This is the biggest term in the polynomial equation, and it can either balance x^3 , or it can balance 1. It can't balance against both of them, it turns out. That

So if we have this term large, and it balances against 1, then when lambda's large, an approximation for the root will be $1 / (2\lambda)$. If x equals $1 / (2\lambda)$, and these are the two largest terms in the equation, then these two will cancel, and f will be very close to 0. So that's one root when lambda is very large.

If I balance these other two terms against each other, I'll see there are two other roots, which are plus or minus the square root of 2λ . So let's start with a large lambda instead, and these as initial guesses for the roots-- and let's trace them back to lambda equals 1. When lambda equals 1, we recover the problem we were interested in before.

So here, I start with three different initial guesses, and I trace them back, and I'm actually able

to find all three roots of the initial equation. So there's something close to minus 1 and 1/2, something close to 1/2, and something close to 1. So in one go, I found all three roots using continuation. Is this clear so far? Sam.

STUDENT: Is that one go or is it three separate [INAUDIBLE].

PROFESSOR: That's an excellent question. So I had to initiate this process with three separate guesses, but it turned out in this case, all three guesses converged to three distinct roots in the end. It's one go in the sense that there's one trajectory that I had to follow, one passive lambda as I had to go through, but I had to solve three times to follow each of these three paths.

There's actually no reason to stop at lambda equals 1. I stopped there because that's the solution to the problem. I'm interested in, but there's actually something richer to see here. If I keep decreasing lambda, eventually these two branches combine with each other, and then they jump down, and they follow the original branch that I had that went between minus 1 at lambda equals 0, and this root about minus 1.6 at lambda equals 1.

So the paths that these trajectories follow can be very, very complicated, actually. A little bit unwieldy or unpredictable. But oftentimes, they can result in predictions of all three roots or multiple roots associated with the system of equations. Yeah.

STUDENT: Going back to the first [INAUDIBLE] solve from lambda equals [INAUDIBLE]

PROFESSOR: Zero to one?

STUDENT: Zero to one. So then would you find only one solution there?

PROFESSOR: Yes. So going from 0 to 1, I found one solution. Going from infinity-- 10 is big enough. 10 to 1, I was able to find three. I didn't know beforehand that I'm going to be able to do that. It happened to be the case that we could with this by construction. This is a particular problem where it works out that way. Other questions?

So this is neat. Like we've done before with iterative methods, we turn hard to solve problems into a sequence of easier to solve problems. Good initial guesses converge fast with Newton-Raphson. We have a sequence of better and better initial guesses, which all converge very quickly to eventually, the solution that we're looking for evaluated at lambda equals 1. So this is one way of getting good initial guesses. You have a sequence of good initial guesses. Make sense?

So we're just here. We had this polynomial problem and we injected λ in a particular place. We injected it there because I knew it was going to work. I knew the problem could be transformed in this way in order to work out and distinguish this. It's not always obvious how to transform a problem to make it look this way, but oftentimes we have physical problems in which there's a parameter that we can do continuation with that is obvious.

So think about fluid mechanics problems. Those are nonlinear partial differential equations. So we're trying to solve say, the Navier-Stokes equations. Rather than jump in and try to solve that right away, we might do continuation.

We might find a solution at very small Reynolds numbers, where the equations are almost linear, exact solutions are known in that case, and we might progressively step the Reynolds number up, and use our solution set small Reynolds numbers as initial guesses for increasingly large Reynolds numbers until we get to the one that we're interested in.

You can imagine doing that in mass transfer problems where you vary the Peclet number. In multicomponent phase equilibria problems, this might happen by varying temperature or pressure until we get to the temperature or pressure that we're interested in. If we have reaction equilibria problems, we might vary the different reaction rates until we get to the right balance of reaction rates to determine the equilibrium compositions.

So they're often natural parameters in the problem that we're free to adjust. We can transform from easy to solve problems to hard to solve problems in a continuous fashion. It's not always going to work out the way that I showed. It's not always going to be so nice and neat. It may be quite a bit more complicated, and sometimes there are problems for which there isn't a natural parameter available to vary.

In those cases, there's another strategy one can employ. It's called homotopy. This is the transformation from one problem into another problem in a continuous fashion. So oftentimes, we're seeking a set of roots, x^* , which are a function of a parameter λ , and they're the roots of an equation that looks like this. So they're the roots of this function, h , which is a linear superposition of a function f and a function g .

When λ equals 0, h is equal to g , and so the roots of h are the roots of g . And when λ equals 1, h is equal to f , and so the roots of h are the roots of f . So we might transform-- we make λ continuously between 0 and 1 in a construction like this. So

we might transform the roots from the roots of g into the roots of f . Maybe the roots of g are easy to find, but the roots of f are hard for us to find.

There's a smooth transformation though, from g to f . When we have those sorts of smooth transformations, there's a chance things will be well-behaved, or at least where we encounter bad behavior-- and I'll show you what I mean by bad behavior in a minute-- there's a reliable way to interpret what's going on there.

So we vary λ in small increments from 0 to 1, and the solution x^* for some λ_{i-1} is used as the initial guess to find the roots at some λ_i , which is a little bit bigger than λ_{i-1} -- our next iterate in λ . Does this idea make sense? Where do f and g come from?

So this is a big question-- f is the problem you want to solve, typically, and g is some auxiliary problem. It can be arbitrarily chosen. There's nothing to tell you why one g should be better than another one, but oftentimes, we choose them in a physical way. So there will be a physical connection between the f problem and the g problem, and I'll show you that right now.

So here's an example. We looked at this example before. We want to find the roots of the van der Waals equation of state. It's written in terms of the reduced pressure, the reduced molar volume, and the reduced temperature, and let's find those roots for a given pressure and temperature.

So this is a function f of the molar volume equals 0, and we're looking for these three roots here. There's a root down here, which is the molar volume is low. Not a lot of volume per number of moles, so maybe this is a liquid like root. Out here, the molar volume is high. We have a lot of volume, the number of moles and materials. This is a vapor like root, and then there's an unstable root in between them.

So three roots to find. We'd like to be able to find them in one go, or we'd like to have some reliable mechanisms for finding them. Homotopy is one way to do this. So let's construct a new function h of the molar volume, which is λf . We want to find the roots of $f + 1 - \lambda$ times g . And as g , let's use something physical. So let's let g be a function that represents the ideal gas equation of state.

So PV is nRT , but I've written everything in terms of the reduced pressure, molar volume, and temperature of the van der Waals equation of states. So you pick up this factor of $8/3$ there.

But this is the ideal gas equation of state. Its roots will be the roots of the ideal gas equation of states. I'm going to transform from something that's easy to solve.

I know how to find the molar volume here, there's just one, to something that's hard for us to solve. It has three roots-- the roots of them van der Waals equation of state. So when λ equals 0, I'll get the roots associated with the ideal gas, and then λ equals 1, I'll get the roots associated with the van der Waals equation. Yes?

STUDENT: If you're starting with λ as 0 and you get the roots of g , you only get one root, but you're trying to find two roots [INAUDIBLE]. So how does that work?

PROFESSOR: I'll show you. Here's what the process might look like. I've got to start with a reduced temperature and pressure. Here's my initial guess for the molar volume. It's the root of our ideal gas equation. Here's the definition of f , and the definition of g , and the definition of h .

l times f plus 1 minus l times g , and I'm going to loop over all of my λ s, solve the equation subject to some initial guess, and update my initial guess every time I update λ . That's fine. That's the code you. Can implement the code if you want or play around with it.

So what happens if I try to carry this out? So I'm going to vary λ between 0 and 1. I start with my initial ideal gas guess for the root, and as λ grows to 1, the molar volume shrinks until I get to λ equals 1, and I've found one root of my equation. One root, good. There's not necessarily any reason to stop at λ equals 1.

That's the root I wanted, but there's no reason I have to stop there. So I could continue. If I keep pushing λ up higher and higher, eventually I'll get to a point where all of a sudden the molar volume jumps down to a different place. This is the result of the algorithm. I just told λ to go bigger. But there will be a discontinuity or a cusp here that I jump off of, and I find myself on a different solution branch, which is sort of crazy.

So now I can do something clever. I found myself on a different solution branch. Why not try decreasing λ along that solution branch instead? Maybe I'll find a different solution, and it turns out that's what happens. All right, so I jump down to this other solution branch, and now I try decreasing λ from 2 back down to 1, and sure enough, I found the vapor molar volume, and now I found the liquid molar volume.

There's no reason to stop and λ equals 1. So if I keep decreasing λ , eventually I'll

hit another cusp, and I'll jump back up to the original solution branch. So I do my homotopy. I vary λ from 0 to 1, and I find one root, but if I keep going, I might identify a cusp, and jump off that cross. I might reverse my direction with the homotopy, and I could find another root. I might hit another cusp as well.

Now, when I hit a cusp, maybe it's best not to jump off the cusp, but instead to reverse direction at the cusp, and try to trace it out the other direction. And if I do that, if I get right to this last point, and I try to reverse direction again, I can trace out one more solution path, and I can find the third root to the equation.

So in one go, if I'm paying attention to how my roots are changing-- if they change by too much, I might switch the direction with which I change λ , and I could find all of these roots at once. So three roots in one go. Yes.

STUDENT: Is this like trial and error? How do you know when to change?

PROFESSOR: This is a wonderful question. So in this case, you can do it by trial and error. You can detect these jumps. That's even better. You look for jumps in the value of the solution that are of sufficient magnitude, and when you detect them, you can reverse the direction.

There's an even more systematic way to do this, which is called arclength continuation. Give me one second, and I'll give you slide on arclength continuation, which is a systematic way of doing exactly this process of tracing out this path in the solution plane. Roots versus homotopy parameter.

These cusps here, are sometimes called turning points, and they have a special property, which is the determinant of the Jacobian of the homotopy-- the homotopy function-- is going to be equal to 0. The Jacobian is going to be singular right at these cusps. So you can detect them by tracking what the Jacobian is doing right.

The Jacobian is going to be non-singular at most of these other points, but when you hit one of these turning points, the Jacobian will be singular. There isn't a well-defined direction for the homotopy to proceed along. It hits this cusp where the slope is infinite in this plane. There isn't a well-defined direction to step along here.

The question was-- OK, it seems a little ad-hoc. You've got this 2D plane, and it's easy to guess when you jump from one solution to another, and trace the path back. That's fine.

There's a systematic way of doing this, which is called arclength continuation, which says I know that there's some curve in this solution plane. There is some curve here. I've shown it to you graphically.

Let's try to parametrize that curve in terms of a measure of the arclength. Call it s , the distance along the curve. So the roots are a function of λ , which can be a function of the distance along the curve, and λ is a function of the distance along the curve, and there is a constraint which says s has to satisfy an arclength formula.

This is the formula for the arclength in this plane of roots versus λ . The derivative of this function with respect to s squared plus the derivative of λ with respect to s squared equals 1. That defines s as a measure of length along this curve. You learned arclength formulas in the context of integration at one point. This is the differential version of that same formula.

So I've got a differential equation. Well, it's actually a differential algebraic equation. This is an algebraic constraint that needs to be satisfied for all values of s , and it depends on how the roots change with s . We'll learn how to solve differential algebraic equations in the two sections from now.

So we don't quite know how to solve something like this yet, but if we know how to solve those sorts of equations, then it's clear that we can trace out this entire curve in the solution plane, and then go back and try to find the places where the curve intersects λ equals 1. So if we follow this path by solving this equation from s equals 0 out to some very large s , then we should be able to find all of the roots that are connected to the path. Yes.

STUDENT: If there's a turning point that [INAUDIBLE] then does that imply multiplicity?

PROFESSOR: Yes. So it's a real problem. That turning point singularity is actually built into this arclength formula as well. This is the derivative of x with respect to s squared, which takes the direction you're moving in out of the problem. This is a length, not a direction.

So when we get to these turning points, the solution methods for these equations are going to have to know that I was headed in a downward direction first. I can get to this turning point, and then the question is which direction do I move in from the turning point? Do I go up or do I go down?

Which way does the solution path change? The slope isn't defined there, so it's quite

challenging to figure out exactly how to proceed. But it is what you say. It's like the solution has multiplicity at that point. Curves are tangent in the solution plane. It's a great question.

OK, so that's the notion of arclength continuation. We may talk about this later when we talk about differential algebraic equations, and we'll talk about some equivalent sets of equations that we know how to solve, but they have to satisfy this constraint that s is a measure of arclength along the curve. But there you go. You can find all the solutions, at least all of the solutions connected to this path defined by the homotopy equation.

You can find all of them in one go using some clever methodologies. Oftentimes, we're not just worried about these turning points or cusps, we encounter bifurcations in the solution. So we may have a path that has one root along it, and then that path may split, and there may be more than one root. Here's a simple example.

Find the real roots of $x^3 - rx$. If r is negative, there will always be one real root, and it'll be $x = 0$. If r is equal to 0, there will still be one real root, it's 0. That root will have multiplicity 3 in that case. And if r becomes positive, I'll suddenly have three real roots instead of one. So as I vary the parameter r from a negative number to a positive number, I'll go from one real root to three.

In the solution plane, that will look like this. Here are the roots as a function of r . r is less than 0. All the roots are equal to 0. When I hit $r = 0$, I have multiplicity associated with the root, and the solution bifurcates, and there will be three possible solutions now.

If I was trying to do a homotopy problem where I vary r continuously from some negative number to some positive number, I'll hit this point, and I'll want to follow all three paths to find solutions out here at positive r . But I can detect those points.

Just like I was said before, at these weird turning points or bifurcations where we have multiplicity in the roots, the Jacobian in is going to be singular. The Jacobian of the homotopy is going to be singular. Its determinant will be equal to 0. So I can detect these points by checking the determinant of the Jacobian. And when I find one, I want to do something smart to try to follow these paths.

Your homework this week has a problem like that. You're trying to find all the azeotropes of this ternary vapor liquid equilibrium system, and you'll have a homotopy parameter that you change, and it'll bifurcate. OK, so as you change the parameter, you'll go from one path to

two, and then those two paths will split again.

You'll be splitting from a single component solution into a two component azeotrope, into a three component azeotrope. And you'll want to detect where these bifurcations occur, and try to follow solution branches off of those bifurcations, and there's some good hints in the problem for how to do that.

So occasionally, we'll encounter problems that switch from having one solution to having many solutions. You've seen how this happens in the discontinuous sense with these turning points. Here, for a given value of λ , all of a sudden it's clear that not only is there one solution here, but there's another solution down there.

My path has to choose. Do I want this solution or the one below? I jump off this turning point. That's a sort of bifurcation. These are continuous bifurcations. They're often easier to take care of. So the bifurcations in a homotopy let us find multiple solutions.

Every time we detect a bifurcation, if we just track those solution branches, we split our algorithm up to track each of the branches separately, they'll terminate at different roots to the homotopy equation. And when we get to the value of λ equals 1 in our homotopy, we're there. We found multiple roots to the original equation. These are often of great physical interest.

Do you know what sort of things these bifurcations represent? One occurs in the van der Waals equation. So if I start at very high temperatures and a given pressure, the van der Waals equation will have how many real roots? One-- a gas.

And then as I turn the temperature down, there is a propensity for liquefaction of that gas phase. The formation of a liquid in coexistence with the vapor. So I'll go from having one root to having three roots all of a sudden, and there will be this bifurcation point where that transition happens in a continuous fashion.

STUDENT: Can I ask you a question?

PROFESSOR: Yes.

STUDENT: [INAUDIBLE]

PROFESSOR: Yeah.

STUDENT: So as you early put it out that the turning point, the determinant of the Jacobian [INAUDIBLE] including lambda is singular. But here you're writing j of x . Is that [INAUDIBLE]

PROFESSOR: Excuse me. This is a typo. So there should be a little h here. This is the determinant of the Jacobian of the homotopy equation at that root. Where there's a bifurcation, the Jacobian will be singular. I apologize. That's a little bit--

STUDENT: And that lambda.

PROFESSOR: And that lambda.

STUDENT: Just to make it clear, you have n state variables. Like x is how many unknowns you have, and then you're adding another unknown to lambda, another parameter. This Jacobian has n plus 1 dimension. This is also [INAUDIBLE] Is this correct?

PROFESSOR: Well, no. Actually, the Jacobian with respect to x only is also singular at these turning and bifurcation points.

STUDENT: [INAUDIBLE] as well.

PROFESSOR: Yes. It's also true what you said, that if I have the Jacobian, and I take the derivatives with respect to lambda, that will also be singular, but it will be singular because the subspace, which is the derivatives with respect to x , gives you a singular component.

The condition is the same as here. So it's the Jacobian of h , taking the derivatives with respect to x -- all the different x 's of h . The determinant of that matrix at a particular lambda will be equal to 0, and the value of lambda will be the lambda at which you have this turning point, or which you have a bifurcation.

Here's what it looks like in a two-dimensional sense. So I'll have my homotopy equation where I'm trying to find the roots of h . Say it's a two-dimensional-- x is dimensions. So it has x_1 and x_2 , and these curves represent h_1 equals 0 and h_2 equals 0, and they intersect at one point for a given value of lambda. There's the root.

Now I increase the value of lambda. The homotopy equations change. So these curves change shape, and all of a sudden I go from crossing curves to tangent curves, and we know tangent curves in the plane will have a Jacobian which is singular. And this is the bifurcation point.

These two curves become tangent, and as I go to a bigger value of λ , they'll change shape again, and I'll now have multiple roots. I'll bifurcate from one solution to two, or one solution to three, but the bifurcation will occur through this transition where I have tangent curves, and we know there, the determinant of the Jacobian is 0. There's a singularity under those conditions. Does that makes sense?

OK, so a couple of things about the practice, and I'll give you a simple example you can try out, which I think is an interesting one, if pathological. So in practice, it's hard to hit this bifurcation point exactly. And I'm telling you that you should be detecting these branches. You should follow these branches, but clearly, I've got to know where this point is in order to follow the branches. That's a problem.

If the determinant of the Jacobian is 0 at the bifurcation point, then it's going to change from positive to negative as I cross the bifurcation point with λ . Actually, I don't necessarily have to find the point where the determinant of the Jacobian is equal to 0.

What I really should do is try to track the sign of the determinant of the Jacobian, and see when it goes from positive signed to negative signed. And when it does that, it will have crossed through one of these bifurcation points or one of these cusps.

That's the point at which I should stop, and say, OK, there's possibly other solution branches to follow. So as I step from my smaller λ to my bigger λ , the sign changes. I want to step in some different directions than the direction I was heading before, and those different directions will trace out the other solution branches.

It turns out those different directions belong to the null space of the Jacobian. So if you find vectors that are very close to the null space of the Jacobian, those directions are the directions you should go to find these other solution branches. So it's very interesting. Yeah.

STUDENT: Can you have a Jacobian that goes from positive down to 0 and [INAUDIBLE]?

PROFESSOR: That may be possible. You won't encounter that in the problem that you're doing. That would be a very difficult problem to do branch detection on. Let's Suppose that was the case and you want to find exactly where that bifurcation point is. These bifurcation points are of physical significance.

In the van der Waals equation of state, that bifurcation point is the critical point. It's the critical

temperature at which that phase separates. Maybe I want to know that exactly. Well, all I need to do is solve an augmented system of equations. So let's solve a system of equations, which is our original homotopy equation equal to 0, and determinant of the Jacobian of this homotopy equation equal to 0.

This is if h had dimension n . This is now n plus 1 equations, and let's solve it for n plus 1 unknowns. Let's solve it for x and for λ . x has dimension n , and we add one more unknown-- the value of the homotopy parameter at which this bifurcation occurred. So we can find that point exactly.

So there may be cases where this happens where it doesn't continuously transition from positive to negative. It may hit 0 and go back up, but that's a really challenging case to handle, but you can handle it. You can find precisely the point at which this bifurcation occurs by solving these augmented equations. It's just another non-linear equation to solve. Yes.

STUDENT: I'm not sure about the previous slide. So in the previous slide, why is the [INAUDIBLE]

PROFESSOR: Let me go back to here. So let's let the blue line here, be the place where the first element of h is equal to 0. So this is all the values of x at which the first element of our homotopy top equation is equal to 0. And the red curve here, let's let that be all the values of x at which the second element of h is equal to 0. And so h itself, is only equal in 0 where both of these curves are equal to 0, and that's their intersection.

So they intersect here, and this star is the solution. It is the root of this non-linear equation up here. Now, I increase λ . I make λ a little bit bigger. That changes h . h is a function of λ . So h changes, and these curves change shape in the plane. And at that bifurcation point, these two curves will become tangent to each other.

There will be one root, but in this solution plane, the two curves will be tangent. And we saw before that when we get these sort of tangency conditions or the equivalent in higher dimensions, the Jacobian of our function we're trying to find the roots of will be singular. It's like a multiple root.

And when we increase λ further, curves change shape again. I'm not changing the red one because that's complicated. I just changed the blue one for you as a graphical illustration. So the blue one changes shape, and now I've gone from having one solution to having three. So in order to go from one solution to three, in a continuous fashion, I have to go through this

tangency state, and that defines the bifurcation point. Does that make sense?

Here's a simple example you should try out. So I'll present it to you, and then you should try to solve it in Matlab. You actually know the solution, geometrically, so it's not too hard. So here's a function. The first element to that function, it's a function of two unknowns-- x_1 and x_2 .

The first element to that function is an equation for a circle of radius r with center at minus 3 minus 1, and the second element is an equation for a circle of radius r with center at 2, 2. So there's two circles. We want to find the radius where the circles just touch, and that point is a bifurcation point.

If r is perfect, these two just touch. If r is a little too big, the two circles overlap, and there are two solutions to this problem. And if r is a little too small, there are no solutions to this problem. So the perfect r for touching is a bifurcation point. It has all the properties of a bifurcation point, just like we discussed. It's a multidimensional equation and it bifurcates. You know the solution to this problem, geometrically.

You could work it out in a couple of minutes, but you could also solve it using these augmented equations. So we know that point at which the circles just touch, the value of r at which those circles just touch, will be the place where f of x is equal to 0 at which these equations define a circle, and at which the Jacobian of these equations-- the derivatives of f with respect to x , it's determinant is also equal to 0.

So we want to solve this system of three equations for three unknowns. The values of x_1 and x_2 , where the circles just touch, and the value of r , the radius of these circles at which they just touch. That would find the bifurcation point. That will find the touching of the circles. Can you see that, geometrically?

All these bifurcation problems work exactly that way. There's some homotopy parameter. In this case, that's r . But there's some parameter we're varying as we pass through that bifurcation. We'd like to know the values of our unknowns and the value of this parameter at that bifurcation point or at that critical point.

So you find this point by solving the augmented equations. We have a non-linear equation-- three nonlinear equations for three unknowns, and you can do that with a Newton-Raphson iteration. It's a non-linear equation. We know Newton-Raphson is the way to solve these things. To do Newton-Raphson, that means we need to compute the Jacobian-- this is

confusing.

We've got to compute the Jacobian of these augmented equations, the derivatives of these augmented equations with respect to our augmented unknowns. So that will be the derivatives of f with respect to x , the derivatives of f with respect to r , the derivatives of the determinant of the Jacobian with respect to x . That's this gradient here.

The derivatives of the determinant of the Jacobian with respect to r . This is our augmented Jacobian. The augmented Jacobian times the Newton-Raphson step. That's equal to minus the function we're trying to solve evaluated at the previous iteration.

Solve the system of equations, you'll get your Newton-Raphson step, and you can find the solution to the augmented equations. It'll converge quadratically. You should commit yourself, you can figure this out. Don't try to compute all of these derivatives. That's complicated. You're going to make a mistake. I'm going to make a mistake if I try to do that.

What do we do instead to compute the Jacobian? How do we compute these derivatives? What's that? Finite difference, right. Don't play games with it. Try to use finite difference to compute the elements of this instead. You know the exact solution too, so you can use that as an initial guess, or something close to the exact solution as an initial guess to see if you can find this bifurcation point.

So homotopy bifurcation, there are ways to get good initial guesses by continuously changing our problem. We get a succession of problems that will converge faster because we have good initial guesses, and that will give us a reliable convergence, hopefully, to one of the roots of the equation we're after.

If we're able to pick up solution branches, we can track these cusps, or find these bifurcation points, we might be able to find multiple solutions. There are problems for which depending on your choice of g in this homotopy function, maybe some of the solution branches are disconnected from each other. Boy, that's crazy and complicated.

We're not going to worry about those sorts of cases, but they can pop up and happen. But the point is there are a robust strategies you can employ to find multiple roots of these equations or multiple minima in optimization problems using precisely this technique.

You'll have a chance to practice this on this week's homework assignment, and I think you should take a look at this problem. It's simple. It looks complicated, but if you can understand

this from, then you'll understand everything you need to know about bifurcation and homotopy. Any questions? Yes.

STUDENT: At a bifurcated point, we go from having one root to three roots, per se. And you said that when that occurs, we need to follow each of these solutions, right?

PROFESSOR: Yes.

STUDENT: And you said that we need to step in a different direction. So what does that exactly mean? In a 2D sense, we have this stepping forward or stepping backwards.

PROFESSOR: This is a wonderful question. So let me go backwards here. Here's the bifurcation point, and when we go through that bifurcation point, there are now going to be three solutions. And I'm going to need some initial conditions for my solver to find each of these three solutions.

I can't give them all the same initial condition. I'm going to try to track all three of these branches with different initial conditions. If I give them all the same initial condition, and my algorithm meets the minimum threshold for what an algorithm should be, they're all going to converge to the same root. So that's useless. So I need to give them different initial conditions.

So what are those supposed to be? In your problem, I'll give you explicit directions for how to find those. It turns out those initial guesses should be perturbed in directions that belong to the null space of the Jacobian. So I want to perturb from that one solution, which was here, in a direction that's parallel to the tangency of those curves.

Those two curves were tangent with each other. In the other solutions, they're going to sit off in the direction that's tangent to those curves. So I've got to find the vector that's in the null space, or very, very close to the null space, and that will tell me how to get better initial guesses for picking up these other solution branches.

It's not trivial. It's actually quite challenging. You'll have a chance to practice. It's part five of five parts on this problem. The first four, they're relatively straightforward and easy. Part five is a little bit tricky. You get explicit directions, and the TAs know that that's the most difficult part of that problem. So we'll help you through it. It's something interesting to think about trying to do.

You may find you're only able to pick up some of these solution branches, but not all of them. That's OK. There are ways to pick them all up if your algorithm is designed well enough. You

may just pick up some of them, and that may be the result you turn in. You can look at the solution to figure out how the other ones are picked up as well.

But you're looking at the direction along these tangent curves. That's a vector in the null space of the Jacobian. And if you perturb in those directions, then you can trace out these other roots so you can find the bifurcation. It's a wonderful question. You'll get a chance to practice on it. OK, last question.

STUDENT: [INAUDIBLE]

PROFESSOR: Yes.

STUDENT: Is this supposed to be J_{rx} ?

PROFESSOR: That's the Jacobian of f . That's the derivatives of f with respect to x .

STUDENT: Not including r ?

PROFESSOR: Yes. Not including derivatives with respect to r . Just derivatives with respect to x . So this is the Jacobian of f . The derivatives of f with respect to x . The determinant of that will be 0 at the bifurcation point. It'll turn out that-- this will be the last thing and then I have to get off the stage here. It'll turn out that this matrix is also singular at the bifurcation point.

It will be singular because the subspace J is singular. So it'll turn out to be singular as well. Don't worry about it. The condition for the bifurcation point is that the determinant of the Jacobian is equal to 0, and that point is a root of the equations you're looking at. OK, thank you.