

IntroToLinearDynamicalSystems-Lecture02

Instructor (Stephen Boyd): Looks like we've started. Let me start by making – there were a couple of announcements I should have made last time and forgot to. Actually, some of it involves stuff we didn't know then. The first is we have to apologize. The readers that you have – it's a very minor error. The homework exercise numbers after lecture ten have the wrong numbering. We apologize for that. We caught this a week ago, silently updated all the PDF files on the website and had the readers not been made would, of course, deny that we ever did this. But with all the readers out there, we can hardly deny it. All the materials there are in the same order. It's just the numbering gets weird.

The second announcement I wanted to make is for contacting me or the TAs. Please use the staff email address. It's on the webpage. That goes to all of us, and that's a very good way – that way we can see which emails have been responded to and which have not. Please do not email the TAs' personal email addresses. That means that other people in the teaching staff haven't seen your email. Please use the staff address.

I think I did ask if people had seen mat lab, and it seemed like a lot of people had. That's great. I do want to say that in fact you are perfectly welcome to use octave. We'll do whatever it takes, which should be, theoretically, nothing, to make anything that you have to do in this class run under octave. The difference is octave, of course, is a new open source thing. Mat lab runs on a bunch of Stanford computers. By all means, please do not go out and buy mat lab for this class.

You're also welcome, by the way, if you're crazy enough or whatever, you're also welcome to use python or something else. Jacob will help you with that. Is there anybody who is tempted? By the way, the "programming" you'll be doing in this class will be limited to one screenful. We're talking 15 lines at most. Any python – really? No one's gonna take us up on that? Usually there's one person. We never hear from them again, of course.

There was one more thing I wanted to say. The last thing – I wanted to say a little bit about how this class compares with CME 200. You may never have heard of CME 200. You should know about it. It is a class. It's taught also this quarter. If you were to compare the material listed in the abstract for that course and this one, you would see a bunch of overlap, enough to make you have some questions. I think I can explain a little bit about what the difference is.

CME 200 actually focuses on computation. It actually focuses on – it's a good compliment. The classes are meant to be complimentary. CME 200 will tell you how to actually carry out the numerical calculations. This class will focus more on modeling and applications of the materials. They're complimentary. Is there anyone here taking CME 200? It goes the other way in the other class.

Now some administrative things. We have progress. The first is we have a section. That's good. Let me repeat that the most valid information you'll always find on the website, but now that we have a section, which is going to be Mondays 4:14 to 5:05, it will be televised. That's the section, and that sets the whole timing or phasing diagram for the whole class.

That says that the homework will actually be due Thursdays, and that includes homework one. It's true, we did at one point say homework one was due on Friday. We changed our mind. It's going to be next Thursday, a week from now, so we don't feel guilty pulling the homework due date up one day. The homework will be due Thursdays.

The section is on Monday, and then immediately following section, there will be TA office hours, and then on both Tuesday and Wednesday, there will be TA office hours from 4:00 to 6:00. Where is the section? It's Gates B3. Since Jacob said that, it's likely to be correct.

Another thing is that we coordinated with some of the other classes many of you are taking and we worked out a tentative date. Let's just say it's the dates for the midterm. The midterm is gonna be either October 26 or 27 or 27-28. That's your choice. In other words, midterm and final will be something like that. Each one will be 24 take-homes and you'll have a choice of two days, so whichever fits better in your schedule.

I should also mention that if you have a moderately good reason why neither of these dates works for you – it doesn't have to be a superb reason or anything like that, but if it's a moderately good one or it's funny or something like that, we'll make arrangements for you to take it another time, generally speaking beforehand. That means you become a beta tester. You might want to mark these dates in your calendar.

The final is going to be – this is to protect my legally so the registrar's police unit doesn't come pick me up. This would be the final, but you can't hear in my voice the quotes. If anyone asks you, it's not the final. I think it's the last homework assignment and it just counts for a lot. Actually, I think that's illegal, too, because you're not supposed to assign homework on the last day of class. I'll figure something out. I should probably put in my cell phone an attorney's number so when I do get arrested by the Stanford registrar's office I know who to call.

That's traditionally the last day of class. These things usually go out at 5:00 in the afternoon and then people come back a day later, tired but alive. We haven't lost anyone yet. That's gonna be the last day. Once again, if these don't work for you, we can work something out.

By the way, if there's some weird thing where you have to take it after by some significant number of days, that's something you need to let us know really early so that we can work around that. We try to release solutions as soon as possible, but we can't release any solutions until everyone has taken the exam. If there are any quite strange scheduling things, people should get in touch with us soon.

Now I think I have covered all of the administrative stuff, including what I forgot to mention last lecture. Sorry. Today and for the next week or so, what we're covering should be review. What I'm mainly going to focus on – if two weeks from now you find that we haven't covered anything that you don't know, don't worry. We will.

What I want to focus on is really the meaning of all this, because that's something that's conspicuously absent in your traditional linear algebra class. Last time, we looked at the definition of a linear function, a function that takes its argument and N vector, returns an M vector is linear if you can form a linear combination of the inputs and apply the function, and it's the same as if you had applied the function to each of the arguments separately and formed the same linear combination.

This formula, which looks very innocent, in fact involves – it says a lot and it involves a lot of overloading. What I mean by that is that all sorts of symbols are doing multiple duty here. This plus is technically not the same as that plus. This αX is not the same as this αX . Those are basically in different dimensions and things like that. There's a lot of overloading there.

At the very end of the last class, we got to the matrix multiplication function. It's a very important function. Very simple. It's two ASCII characters. It's this. You've taken argument X and you multiply a vector and you multiply it by matrix A and that's what you return. That's the matrix multiplication function. It's linear. You just have to check these things. It's easy to do.

It turns out actually it's more than that's just an example of a linear function. That is actually the generic form. All linear functions can be represented as F of X equals AX . People make a big deal about this. It's actually not that big a deal, although it's useful to think about how given F – if F were a black box or just a function or operator which you could call but cannot see the internals of, the question is how would you reconstruct A , a model? That's essentially the question. That's why you're going to be seeing F of X equals AX a lot.

Now let me say something broad about what does Y equals AX mean. It turns out there's lots and lots of applications of this. In fact, really much of the course is going to be about banging practical problems into a form like this or a variation on it that might be more complicated, but something like that. Here, there are huge, broad categories of interpretation. One is this. In one interpretation, X is a vector of unknowns or parameters that you want to estimate – things you don't know.

Y is a vector of measurements or things you do know or, in communications, this would be something like a transmitted signal and Y is a received signal. That's one very broad category of linear functions. In this case, there will be all sorts of interesting things you want to do. The things you want to do would be something like this.

Suppose you're given Y . You'd like to say as much as you can about X , ideally saying what X is. That would be undoing this linear operation. In another broad category, X

represents an input or action. In this interpretation, X is something we can mess with. It's something we can change. X could be all sorts of things. It could be a signal we transmit. Then this Y is an outcome in this interpretation.

X could be all sorts of things. It could be the thrust that you put on your engines. That could be X_1 . X_2 could be the deflection of your [inaudible] and your elevators and your control surfaces and things like this. So in that case, it's an action. The general idea is we control it. Y is now a result. Y might be a vector of climb rates and things like that. We'll see more specific examples soon.

Another very useful interpretation is that you think of this abstractly as a function or a transformation. It's just something that takes a vector, does something and gives you another vector. An example of this would be something like a Fourier transform, or it could be a rotation. Something that takes a vector, rotates it 27 degrees clockwise around some axis. That would be – there, you'd be thinking in terms of a transform. They're all going to look like Y equals AX .

What I'm about to say is very simple and totally obvious, but I think it still needs to be said. Here's Y equals AX written out component by component. This is the – I'm going to call Y the output and X the input. I'm neutral on whether it's an input we can mess with or it's some parameter we want to estimate or something like that. I'm just – input and output are just neutral here. This says that the I output is a sum over all inputs. It's weighted by A_{IJ} . That's what it says. What that means is you can interpret A_{IJ} immediately. It's the gain factor from the J input to the I output.

Again, what I'm saying is not exactly complicated, but it's very important to understand. A lot of these are very important things to know. That means that the I row of a matrix – that's A_{I1} , A_{I2} , A_{I3} – you're going across the row. All of those things concern the I output. Those are the gains from the inputs to the I output when you scan across a row. If you look at a column of a matrix in Y equals AX , a column concerns an input. So the J column – the third column concerns the third input. In fact, as you scan down that column, what these are is these are the gains from the third input to all the outputs. That's the idea.

That's fine to say, but now what it means is when you see a matrix and you see something in it, you should interpret things immediately. So for example, if you see that A_{27} equals zero – that's the 27 entry in the matrix. It's zero. That has a meaning, and the meaning is the second output, Y_2 , doesn't depend at all on the seventh input. Again, this is obvious, but it needs to be said.

If, for example, you see that the 31 coefficient to the matrix is much bigger than all other entries in that row, that has a very specific meaning. What it says is that Y_3 depends mainly on X_1 . I'm assuming here that the coefficients in X_1 are scaled, so they're all about the same. You could have something weird where the X_1 entries are on order of units and the X_3 entries have numbers like 10 to the minus 9, but I'm assuming they're all about the same. Then you could say this.

You can see all sorts of other stuff. For example, this goes on and on. If you have a column – if you find that A_{52} is a lot bigger than A_{12} , that tells you that the second input affects mainly Y_5 among the outputs. You can just go on and on and on. For example, if A is lower triangular, that means A_{IJ} is zero for I less than J . It means that Y_I depends only on X_1 up to X_I .

By the way, this means – when you have lower triangularity, you're going to want to make a vague symbolic link in your mind to causality. If the index here represented time, this would be something like causality, and that's what lower triangularity means. You shouldn't just dispassionately look at a lower triangular matrix and say yeah, it looks pretty. It has a very specific meaning, and it talks about what inputs affect what outputs and it's very specific.

Here's a very special case. Let's say the matrix is diagonal. If it's diagonal, what that means is that the I output only depends on the I input. In general, you have the idea of a sparsity pattern of a matrix. The sparsity pattern is the list of zero and non-zero entries. In a lot of cases, a matrix has no interesting sparsity pattern.

All the entries are non-zero, in which case you call it a dense matrix. In a lot of interesting cases, a matrix will have a non-trivial sparsity pattern. So blocks of entries will be zero, and that will have a meaning. You should never, ever look at a matrix that has blocks of zeros and not think what does that mean? A lot of this is simple, but this is the idea.

How would you find, for example, if I gave you Y equals AX and I asked you something like this – which input makes the greatest contribution to the seventh output, what would you do? Yeah, so you'd look at the seventh row, you'd scan along it and if the entries were all about the same, you'd say they all affect it about the same. If one stood out, you'd say it effected.

What does the sign mean? What does it mean to say that A_{35} is positive? It means the following. If A_{35} is positive, what it means is that if you increase X_5 , the fifth input, the third output will increase. These are obvious things. Don't think about them too much, because there's not that much here to think about. When we see it in action, it'll make more sense.

Now what I'm going to do is go through a bunch of examples to give you a rough idea – just so you have some pictures in your mind of this thing. So the first one is from mechanical engineering. We look at a linear elastic structure. So I have a structure like the steel frame of a building or something like that. Here, X_J is gonna be an external force applied to this structure at some point and in some direction. It could also be a torque. For example, these could be all the various forces here.

Here I have the first four forces – we can imagine these are wind loadings on the building. These are wind loadings. These might be dead loading in the building. Y_I is an output. That's going to be the deflection of a point in a building in a certain direction. It

might be the deflection of this floor here. It has to be oriented, so for example, I might say that downward is positive deflection. Upward is negative.

There are names for these. If you get into civil engineering or structural engineering, the name, for example, for the difference between the horizontal displacement from one story to the next is called interstory drift. Any time you have some specific area, you'll find all sorts of colorful language for it. We'll say Y is one of these deflections.

It's a basic fact. It's roughly true. It says that if these forces are small – of course, in a steel frame building, small does not mean five Newtons. It can mean quite a lot. If they're small, which depends on the application, then it says that the deflections are approximately a linear function of the applied forces. That's actually quite interesting. Now I can ask you some questions. I have to show some displacement.

I'm going to make the displacement here Y_1 , the displacement here Y_2 and so on. That's Y_4 . Now I'm going to ask you to tell me about some entries of the matrix. Tell me what you can say about A_{11} . It's positive. What's your intuition behind that? You push here with a unit of force. The building will deflect this way, and then whatever the ultimate equilibrium deflection is – it looks like Y_1 . By the way, what can you say about A_{21} compared to A_{11} ? It's smaller. In fact – what sign do you imagine it has? Positive. Okay.

By the way, only those of you who have had a class on structures are actually allowed to say that. The rest of you, like me, have to say we guess it's positive. Tell me about the column that associates with X_3 . Tell me about the third column of the matrix. What do you think it might look like? So Y_3 would be big. How about Y_1 and Y_2 ? I think they'd also be big, because you'd shift it over and there'd be nothing up there. You get the idea.

By the way, the units of this matrix – in general, it's sometimes useful to think of the units – the units here are in meters per Newton if those are the units I'm using for displacement and force. It's got a name. It's called the compliance matrix. If the deflections and the forces are measured at exactly the same place, then the inverse, because the matrix is actually square – the inverse is called a stiffness matrix for example. That's just one simple example here.

Second one is we have a rigid body. For example, let's say a satellite or something like that. Here's the center of gravity. Let's say that I'm going to apply forces and torques on it. I'll apply a torque here. X_1 is a force in a fixed direction. X_1 measures the size of it. X_1 negative means that I have a force in the direction opposite to whatever my reference direction is. X_j is gonna be my vector of these external forces and torques. What I'll do is if I give you the forces and torques, there is a net force and a net torque applied to this rigid body. In fact, those are linear. You have Y equals AX .

A depends on geometry. You don't actually often write it this way. All of that is obscuring the fact that the net moments and torque on this rigid body is a linear function of these applied functions. This comes up. This would be, for example, a vehicle. These could be thrusters or a control surface – anything that creates a force or torque on the

vehicle. This would be a very interesting matrix because it would actually tell you how what you do maps into the total force of torque. There's lots of questions you can imagine you might want to answer here.

So for example, here, the J column of A is a pattern, and it tells you the first three components in the J column give you – let's imagine the J column is a force applied. Then the first three components – A13, A23, A33 – what are the units of those? If the third input actually is a force, what's the unit of these? These are unitless, and the reason is they map Newtons to Newtons. These are unitless. How about A43, A53 and A63? What are they? Meters. They have units of meters because they map Newtons to Newton meters to torques.

If you want to know if that is an actuator or a thruster and you want to know what does the thruster do, you get a rough idea by scanning that column. Looking at the geometry here, tell me something about the second column that corresponds to this input. Let's go right down the second column. What can you say about A12? That's gonna be the X component of the net force contributed by this thruster. What is it? The magnetite is less than one.

Let's be more specific. Zero. Is it really zero? It's not perpendicular. It tilts a little bit to the right. So it's small and it's positive. Let's go for a number. Let's say it's .15. It's very important to know that the number I made up was not arbitrary. If I'm off, I'm not off by too much. How about A22? It's large. In fact, how large? It's close to one. It's .85, .95. Something like that. That's enough on that.

Let's look at a linear static circuit. Here, I have a linear circuit. It's an amplifier, but it doesn't really matter what it is. It's got linear resistors here. It's got an external current source here, an external voltage source here and it's got a dependent current source here. That's how that works. That's a current controlled current source here. Here, I'm going to let X be the value of the independent sources. In this case, that's X1 and X2. Y is going to be any circuit variable. For example, it could be the voltage across a leg. It could be the current through a device. It could be the potential – the voltage at a point.

It doesn't matter what it is. In this case, it's definitely linear. You have Y equals AX here. You have Y equals AX and that's interesting. By the way, depending on what YI is and XJ, all of these entries have interesting units and names. For example, if the Xs are all currents and the Ys are voltages, then A is called the impedance or resistance matrix. In this circuit – this is again only for people in electrical engineering.

What is A11? A11 has a meaning here. I want the street name. It's the input resistance of the amplifier. Again, if you're not in EE, don't worry about it, but that's what A11 is. It says when you pump current in here, you develop a voltage here. The gain is A11. It's in ohms and it's the input resistance of the amplifier.

Next one is a simple dynamic system. It's your basic frictionless table here. You have a frictionless table and you have your standard issue one kilogram mass, and what's gonna

happen is this. It's gonna be at a position zero and it will be at rest at T equals zero. What we're going to do is we're going to subject it to a P wise constant force. So for one second, we apply a force, which is X_1 . Then we apply a force X_2 , then X_3 and so on. So now the interpretation of X is like a force program. Program means it's your plan. It's a force. You could call it a force program, a force plan, a force trajectory.

You can call it all sorts of things. That's what it is. That's what X is. We'll make this up to N . There are going to be two outputs or outcomes, and it's simply going to be the final position and the velocity of that mass. The point is that's linear. That's Y equals AX . What is the size of A ? Two by N , because it's got two outputs and N inputs. I think I mentioned this last time, but I do encourage you to read some things on the course website that are very elementary but set up the notation of matrix multiplication. Just scan it.

You'll also find another PDF file there, which has a title like "Crimes Against Matrices." You might want to scan bits and pieces of it. I'll come back to that later. It means you should always at an instant's notice be able to identify the dimension of something. If you can't – if someone can point to something you're doing or looking at and say what's the dimension of that and you don't know, that means – it's very easy with this stuff to sit back and go yeah, sure.

It's very easy to get complacent here, but you should be checking yourself whether you know exactly – that's related to these things. Y equals AX of course only makes sense if the number of columns of A is equal to the dimension of X . If you ever write down Y equals AX and that's not the case, we call this a syntax error, and it's bad. This is not a good thing. Don't do that.

Here you can interpret stuff. The first row gives you the influence of the applied force at different times on the final position. Actually, I'd like to ask you about that row. What does it look like? I don't want the actual number. I just want intuitively to tell me about that row. It's ascending. What you're saying is a Newton applied at the beginning has less of an effect than a Newton applied at the end. If you say that first row increases as you go down the row, for me, that's this way. If you say it's increasing along the row, it has a meaning.

It basically says – what are the units of the entries of the first row? Is it even linear? Yeah, it wouldn't be in the notes if it weren't. By the way, that's not entirely true. There have been things in the notes that were mistakes for three years at a time. Let's just say I'm pretty sure this is okay. If not, you don't have to do the homework problem where you work out A , but it is.

Assuming that it is linear and the entries of the first row are in meters per Newton because they tell you meters of final displacement per Newton applied, and if you say that that row is increasing, it has a meaning. Do you believe it now? The initial velocity is zero here. The initial position is zero. What do you think? It's decreasing, somebody said. It's decreasing. What does that mean? It says that if you apply a Newton for the first

second, you will get more final displacement than you will if you apply a Newton at the last one second. Is that true? Why?

There's no friction here, so what that means is something like this. When you apply a Newton in the first second, you accelerate the mass to one meter per second. There's no friction. It now coasts for N minus one seconds with that velocity you just gave it. That gives you – you cover a lot of distance. In the last time when you apply one Newton in the last one-second period, what happens is in fact you're merely accelerating it from whatever speed it has to one meter per second more. In fact, the total displacement you get is half a meter in that case.

The details of this don't matter. The point is to think about these things. What about the second row of A ? Yeah, the second row of A is going to be constant because if you apply a Newton for one second to a mass on a frictionless table, it has the exact same net effect. You will increase – we can even say what that row is. It's all ones, because if you apply a Newton for one second to a one-kilogram mass, it will be moving at the end one meter per second faster.

These are simple examples, but you should – this is the kind of thing you should do without even thinking about it. We can put a negative force no problem. The matrix A doesn't care and has nothing to do with whether you push to the left or right. When you push to the left or right in your force program, what you're doing when you push to the left or right is that's changing X . In fact, let me be very specific. Here's an X . Let's make N equals 4 and here's my X . 1,0,0,-1. That has a meaning. That means you push one Newton, you coast for two seconds and you pull. What is Y for this, roughly? It's about three.

In fact, it happens to be exactly three. Y is a two vector, so it's first entry is three. What's the second? Zero. What does that mean? This means that the final position is three meters to the right, and the final velocity is zero. So this is, in fact, a force program that will transfer the mass three meters to the right. It will take it from stationary to stationary three meters.

This is all very simple, but this is, for example, a very simple version of exactly what your disk drive head does when you initiate a seek. This is what happens. It moves from track 25 to track 150. It does exactly this. By the way, the force program is not quite this, but it's roughly like this. Later, we'll find out what it really is.

This one is from geophysics. It's gravimeter prospecting. It works like this. This is underground. It could be seawater or include some air. These are little voxels, and these voxels have a density row, and we'll make XJ the excess mass density over the average earth density. I forget what that is. If one of these has gas in it then X is negative. If this has got some really dense rock, X is positive. When you have an array of things at different density and you actually want to know what is the gravitational acceleration here, you get it by integrating all these things and using the gravitational force business.

Mostly, it's going to be pretty [inaudible] as it's going to be pointing down and it will be about 9.8 meters per second squared. It turns out it actually can be deflected. It's very subtle. It's out there in some digit – something like the third or the fourth digit or something like that, but it will point somewhere else. Its magnitude will change. If you're right next to a giant mountain over there, down is now ever so slightly deflected that way. If you're standing over flat ground and there's a giant, giant cavity under there filled with gas, then in fact the same thing might happen.

This would be deflected slightly that way, and the magnitude would be slightly different. What's often measured is the difference of G with the average around there. You calibrate it somewhere and then you move it around, and in fact, this is approximately linear. Here, A is quite complicated. The formula for it is horrendous and involves all sorts of things. It's got positive and negative things. Very complicated. This is going to involve all kinds of crazy things with sines and cosines and distances and all sorts of other stuff. That will just obscure this fact.

Now you can say lots of stuff. You can say, for example, that the J column of that matrix shows you the sensor readings that would be caused by a unit density anomaly at voxel J . That's literally what it means. The row shows the sensitivity of a sensor. If you make a measurement here, it tells you as a function of voxel what the sensitivity is. You'd expect this to be – the matrix wouldn't be sparse, but it might have a lot of small entries. You'd expect that the gravitational anomaly here would have not a who lot to do with voxels that are way, way off over here. There would still be some structure, if not exact zeros in the sparsity pattern.

Next one is a thermal system. Here, I have a system, and I have some sources of heat in the system. These could be heaters where in a control problem, I want the temperature to do something I want, so these are little resistive heaters. Or these could be processors on a multiprocessor chip where they're just operating and then the temperature is whatever it is. Lots of applications. I inject heat at these sources, and XJ will simply be the heat source, and that's measured in watts. I assume that the thermal transport here is linear, so it's via conduction. It could be by convection, but it would be a very simple linear model of convection.

It's not by radiation, which is going to involve these temperature to the fourth terms. Again, if you don't know what I'm talking about, it makes no difference. It's fine. It's linear. There'll be some appropriate boundary conditions, like they'll be isothermal or they'll be isolated or something like that. Then this whole thing will come to some thermal equilibrium temperature distribution and I'm going to let YI be the temperature at a certain point I . That's the idea.

If you look up thermodynamics, you'll find out – people will tell you that's a [inaudible] equation. It's very complicated. PDEs will come out and you'll pretty quickly – you have to do that if you really want to understand this. But in fact after all the PDEs go away and everything else, you will find that, in fact, Y equals AX . The vector of equilibrium temperatures will be equal to a matrix times the vector of the power dissipation in these

things. It's that simple. Let's talk about A . What can you say about the sign of the entries of A ? They're all positive. What does that mean?

It means that if you pump heat into something, at some point, the only thing you can do is increase the temperature everywhere. Here, what can you tell me about A_{41} ? Small. The idea is this – A_{41} is the gain, and it's in degree C per watt from this heat source to that location. It does have an effect, but it's small. That's kind of the idea. You can imagine all sorts of cool things you might want to do with this. These might be things under your control, and you might say find me an X that makes the temperature distribution something I want.

Let's say for some kind of experiment you want a nice, uniform temperature gradient or you want it uniform. AX is about equal to some desired temperature to the extent possible. We'll be able to answer questions like that. Another one would be estimation, which is I give you 57 measured temperatures. I want you to estimate the power being deposited at these five locations. That's an estimation problem. You want to deduce it from the measured temperatures. There are all sorts of things.

Next one is illumination with multiple lamps. Here I have some surface like this and I have a bunch of lamps at these points up here, and what I can do is each lamp has a power X_J . These things – they go down to this patch here and here, I can actually say what A_{IJ} is. Here, I say Y_I is going to be the illumination on patch I , and X_J is the power in lamp J . One thing to notice when you look at matrices and things like this, and you get used to it after awhile, but when you have A sub IJ , I indexes the output for effect. J indexes the input. So really, the pair IJ is really indexed output, input.

That's weird because most people think of – if you just walked up to someone on the street, they would probably index things by input, output. Matrices are organized as output, input. Too bad, that's the way it worked out. You'll get used to it. By the way, these are dummy variables. I could switch them and be sick and just turn it around and make it JI . That will happen. But right now, I'm just trying to stick to a reasonable – they can be anything you like.

Here, the illumination of a lamp on a patch is given by the inverse square – it's one over the square of the distance and then it's multiplied by this cosine factor, which is basically how much of the light is caught by the angle. By the way, if it's all the way over and the light is below here, you get nothing because you're obscured. That's what the max is here.

Once again, the vector of illumination levels is a linear function of the vector of lamp powers, and that tells you it has the form Y equals AX . Again here, A is non-negative. That's clear intuitively, but it's also clear from this. You get a rough idea. For example, if you look at a column of A , what you are looking at is the illumination pattern generated by that lamp. You're looking at the third column. That column gives you the illumination pattern. If you look at a row, you're looking at a patch. You're focusing on a patch and you're asking what the different gains from the different lamps to that patch are.

We'll look at another couple of these. This one is from communications. Here I have N transmitter receiver pairs. The idea is transmitter J wants to transmit to receiver J . Unfortunately inadvertently it also transmits to the other ones. We don't want that. P is going to be the power of the J transmitter. S is going to be the received signal power of the I receiver. Z is going to be the received interference power of the I receiver. G_{IJ} is going to be the path gain from transmitter J to receiver I . That presumably will depend on how far they are apart. It may depend on all sorts of other things in between.

G is non-negative. You have $S = AP$ and $Z = BP$ where $S - A$ is in fact a diagonal matrix where you just take the diagonal part of G . That gives you the vector of signal powers. You take the rest of G - the off diagonal part of G and you shove that into matrix B , and if you multiply that matrix by the power vector, you get the vector of total interference powers. I'm assuming that the interferences are going to add incoherently. The powers are going to add. It's not coherent addition.

Ideally, what you want is you want A to be large and you want B to be small or zero. That means you want this matrix G to have a very strong diagonal and lots of little off diagonal entries. If I asked you questions like this - the third receiver is most susceptible from interference from which transmitter - how do you find that out given G ? The third receiver is most susceptible to interference from which transmitter? What do you do? The answer is you walk across the third row of the matrix G . I guess for you, that's kind of like this.

You walk across the third row of G . The three entry in G is very important. That's actually the gain the transmitter you want to listen to. You look at the other entries, and in the other entries, you look for the largest entry, and that tells you which transmitter you are most susceptible to. These are simple things, but this is the kind of thinking you need to do.

The next one is from economics. It involves things like cost of production. Here you have a bunch of production inputs like materials, parts, and labor. You combine these to make a bunch of products. We'll let X_J be the price per unit of production input J . A_{IJ} is gonna be the units of production input J that you need to manufacture one unit of product I . So that means if you go cars the row of that matrix, it corresponds.

If you go across the third row, it basically tells you how much - what are the inputs you need to make one unit of the third product. If there's a zero there, it says you don't need that. If it is a large entry, it says you need a lot of whatever that is. Could there be a negative number? Strangely, it depends. Generally speaking, no, but in fact, yes, you could have a negative number. What would it mean if A_{23} were equal to minus one? What could it mean? It's a byproduct. Exactly.

It says that when you make one unit of product two, not only will you not need input three, you will actually as a byproduct of making that generate one unit of input three. A lot of these things where normally you think of it as something that's positive, there usually is a really interesting interpretation of what happens when it's negative.

I'll show you an example where that's not the case. Transmitting negative powers – people have tried to do it, but so far, it just hasn't worked out. How about this one? What is it? You could have active cooling. They have these things and yes, you can pump one watt out. I just mention this because it's good to keep these in mind.

If Y is the production cost per unit of product A , you have Y equals AX . This is beautiful. This tells you something like this. This tells you how the cost of making your family of products depends on the vector of input prices. For example, I could ask a question like this. Among all the products you make, which is most sensitive to the price of energy? How would you answer it? You look for the energy column. It says energy is X_3 . You scan down the third column and you look for the biggest entry and you say that product is the one most sensitive to a change in the price of energy.

Let's move on to the next example. These do get a little bit boring, but it means that when we do talk about stuff that's abstract, at least it has meaning in all of these contexts. The next one is from networking. I have N flows in a network. A flow is something that passes from one node across an edge to another node to another node. These are going to have rates F_1 through F_N . It doesn't matter. These could be in bits per second. They could also be, for that matter, in liters per second. It can also be electricity. It could be anything. It could be goods. These could be transported by trucks. These could be packets.

It doesn't matter. They pass from a source node to a destination or some fixed route. The traffic on a link – some of the routes will go over each link. If no route goes over a link, then it's utterly unused. But every link will have some routes go over it. It may be one, two or 100. The total traffic on that link is the sum of the flows of the routes that pass over it. You can write this as exactly the same thing. If you have T , that's the traffic vector, and that is a vector that tells you – its index refers to a link, and it says this is the traffic on all the links. It's a linear function of the flow rates. It looks like AF .

A is, in fact, a very simple matrix. People call it a zero one matrix or something like that. It basically encodes which flows pass over which links. Now, I can ask you a question. A bottleneck is a link that has a large number of flows going over it. If I gave you the matrix A , how would you find a bottleneck? You look for a row, and a row corresponds to an output – in this case, a traffic. A row corresponds to the contribution to a link from all the flows. You look for a row that has a lot of ones in it. There we go. That's what you said, right? You're right. That's it.

By the way, what's the meaning of a column with a lot of ones in it? It's a long – it's a flow with a long route. That's the idea. Interestingly, we can do this. Let's say that each link has a delay on it. In other words, when you're going to go over a certain link, you actually arrive at that node. There's a queuing or transport delay. It doesn't matter. These are just applications to give you some context for all of this. Each link has a delay, and that's the delay it takes – it might be waiting to get queued up and transmitted. It might be the transmission delay. It doesn't matter. It's whatever the delay is across it.

Therefore, if you have D_1 through D_M are the delays on each link. That's the link delay vector. The latency of a flow basically is the sum of the delays along the route. If these were packets, it literally tells you if I inject a packet here how long it will take before it emerges at the other end. That's the latency, and it's simply the sum of the delays along the route. It turns out this is very easy to write down. It's $L = A^T D$ where this is the transpose of A . In other words, you simply take – you can work it out.

There's some very interesting things here. For example, if you work out what $F^T L$ is, F is a vector of flow rates. Let's just say it's in bits per second or packets per second. L is a vector of the same size. For each flow, it tells you the latency, which is basically the delay. When you inject a packet, how long it takes before it emerges at the destination. $F^T L$ is exactly the number of packets in transit. That's what it is. $F^T L$, which is the sum over $F^T L$ over I is exactly the total number of packets in the network. That's what this is. This you can write out lots of different ways.

You can also write it, by the way, as $F^T A^T D$, because $L = A^T D$. This is – just doing some simple matrix arithmetic, I can rewrite this as $F^T A^T D$, but $F^T A^T$ is the traffic. So it turns out it's the same as this. This is an inner product. Let's talk about what this is. What is $F^T A^T D$? It is exactly the number of packets in transit or waiting on link I . This is the sum this way. You get the same – these are two ways to get the total number of packets in the network. You either sum over the flows or you sum over the links and you get the same thing.

By the way, if there's any of this you didn't get, you should go back and make sure you believe it. Don't spend too much time on this because I've had people come back to me later and say yeah, but just the way you were saying it, it sound like it was very deep. Nothing we've said today is complicated or deep. If you think you don't get it, you do. I've had people come and say things like I understand everything, but I think there are subtleties I'm not getting to which I'd respond there are no subtleties. What we covered today was trivial – interesting and important, but trivial.

Here's a generic source of linear mappings. It is linearization, which you've seen. There's another name for it. It's sometimes called calculus. Here it is. I have a function that accepts an N vector and returns an M vector. You say it's differentiable at a point. Whenever X is near X_0 , the function value is very near $F(X_0) + DF(X_0)(X - X_0)$. You have to be careful here. That is gonna be an M by N matrix. This is DF . That's the derivative of F at evaluated X_0 . It's entries are these partial derivatives. Very near is a technical term. If I did this, this is the definition of F being continuous.

It says basically if you're near one point and evaluate F , you're near the image of that point. That's the definition of continuous. The definition of differentiable is this. Very near, by the way, means it says that the error here is like the square of the error here. That's what the very near is. Very, very near means it goes like the cube of the error, by the way, and it goes on from there. You can read this informally or formally. In many cases and in lots of contexts, people focus – what they do is they focus – lots of contexts

have different names for this. For example, in a circuit, X_0 would describe the bias or operating point.

You'd have something like – these would be the bias voltages, the bias inputs or whatever, and then the deviations would be called the small signal values. You've seen that. In aeronautics, you would have the so-called trim condition. The trim condition is that your thrust is at such and such a level. Your elevators are at this level and so on and so forth.

You're in level flight at 40,000 feet at such and such a speed. That's your trim condition. ΔX represents a small change from that and a change in your elevator deflection. ΔY would be a difference in, for example, the net moment in torque on the airframe. Different fields have different names for a base operating condition and then wiggling around it.

A lot of people introduce a notation like ΔY is Y minus Y_0 . ΔX is X minus X_0 , and then you can write it this way, and this basically says the deviations in a response or output is a linear function of the deviations in the input. That's the generic example. You've seen this. That is literally calculus. The problem is all these stupid multivariable calculus classes make everything complicated by bringing up things like radiance and curls and things like that. This material was super useful in the late 19th century and maybe up until the 1930s.

It's not really much anymore. You need a few people to know all those things, but not really. The problem is that if you look back at your multi-variability, it's unbelievably simple. It's just DF . In fact, that is – this is an approximation. Just by syntax, this has to be – this is an N vector. This side – what you're approximating is F of X , which is an M vector, so the only thing you could multiply an N vector by and get an M vector is an M by N matrix. So that's just an M by N matrix. It would be, for example, in the case of the gradient, it's actually a row vector, which is the right way to write it.

This is something like this. This is the derivative of F evaluated at X_0 . That's a matrix and then this is indexed at IJ . That's what the parsing is here. The left and right hand sides of this are numbers. I, J actually are the indices. I index actually the component of F . F returns an N vector, so F_3 or F of X sub three or something represents the third component of it. J indexes into the input.

We'll look at a specific example of this. I won't do that. I'll just wrap up a little bit. The things we looked at again – this is the last day where you'll be subjected to me looking at stupid examples of Y equals AX . On the other hand – we didn't cover any actual material today. We just looked at a bunch of examples. Don't worry. The rest of the class is not going to go this way, but still, it's an important thing to do. I guess we'll wrap up next time. Remember, there is actually a section on Monday.

[End of Audio]

Duration: 81 minutes