HERBERT GROSS: Hi. Welcome to our lecture today in Calculus Revisited. And our topic is going to be the dot product. And by way of a preamble, let me point out that in our development of vector spaces in this block, up until now, we have deliberately not mentioned the concept of the dot product, even though dot product plays a very prominent role in the usual vector algebra of two- and three-dimensional space.

And the reason for this is that we wanted to emphasize the structure of a vector space. In a manner of speaking, mathematical structures are pretty much like an anatomy-type course, that one starts with a basic structure, exploits it, sees what ramifications it has, and then gradually adds on more sophisticated layers of nerves, nerve centers, skin grafts, and what have you. And ultimately, builds up a superstructure based on top of an underlying basic fundamental model.

Now, without going into that in more detail right now, let it suffice to say that I call today's lecture "Dot Products." And by way of review, notice that in the usual two- and three-dimensional case by the dot product of two vectors, we mean a real number. And if we want to keep this thing vague and not indicate the cosine of the angle between two vectors, et cetera, one usually says that a dot product is a mapping that carries ordered pairs of vectors into the real numbers.

This is the mathematical way of talking about what a dot product is without referring to angles or lines. It's a function which maps ordered pairs of vectors into real numbers. In particular, the mapping must have certain properties to be called a dot product. And again, emphasizing mathematical structure based on what we already know is the case from the two- and three-dimensional situation, what properties do we want the dot product to have?

Well, we would like alpha dot beta to equal beta dot alpha, because that's certainly happened in the two- and three-dimensional case. We would like alpha dot beta plus gamma to be alpha dot beta plus alpha dot gamma. And if we have a scalar multiple of alpha dotted with beta, we would like that to be the scalar c times the number alpha dot beta.

Or if we wanted to rewrite this in vector form, that the c could also be used as a scale of multiple of beta. That c alpha dot theta is the same as alpha dot c beta. And where did these three rules come from? They came from the properties that the ordinary dot product has.

By the way, a dot product becomes known as a Euclidean dot product if, in addition to the given three properties, we also know that the dot product of a vector and itself, alpha dot alpha, is a non-negative real number. And that in particular, alpha dot alpha can equal 0, if and only if alpha equals 0. By the way, let me point out that this fourth condition is independent of the other three.

That when one studies vector spaces in more abstract detail, one usually defines a dot product to have just the first three properties that we talked about. And if the fourth property happens to be present, then one talks about a positive-definite dot product. But not every dot product has to be positive definite, except that if we're trying to imitate the distance property of real vector spaces, then we add the positive-definite criterion. And that's explained in more detail in the study guide as we do the exercises.

But I did want to simply point out that once you know that you have a positive-definite form, that alpha dot alpha is non-negative, then it makes sense to abstractly define the norm or the length of alpha to be the positive square root of alpha dot alpha. You see, this makes sense, because you know that alpha dot alpha is a non-negative real number. Hence, it has a unique positive square root. And this is how we abstract the usual distance function.

At any rate, to show you how this works in abstract form, let's take a three-dimensional case. Let's suppose v is a three-dimensional vector space, that a particularly-chosen basis for v is u1, u2, and u3, and let's suppose that we have a positive-definite dot product. And I'll just make this up at random or semi-random here.

Let's suppose that I know what the dot product looks like on all of the pairs of basis elements. In other words, I know what u1 dot u1 is, what u2 dot u1 is, what u3 dot u1 is. And by the way, by the property, that u2 dot u1 must equal u1 dot u2, et cetera, by the commutative property. Notice that once I know what u2 dot u1 is, I also know what u1 dot u2 is. Once I know what u3 dot u1 is, I know what u1 dot u3 is, et cetera.

At any rate, there are nine possible combinations here. And just try to fix these in your mind, because I'll be referring to these as we go along. u1 dot u1 is 3. u1 dot u2 is 4. u1 dot u3 is 5. u2 dot u2 is 6. u2 dot u3 is 7. And u3 dot u3 is 9. I just happen to know that particular amount.

Let's also suppose that I now want to find the dot product of any two vectors in my space, v. My first claim is-- and again, this is an overview. I'll do this in more abstract detail in the exercises in the study guide. My first point, though, is that once we know what the dot product is for each pair of basis vectors, we know what the dot product is for all vectors and pairs of vectors in the space. That in particular, the four rules-- actually, the three rules that we have for defining a dot product tells us how to compute the dot product for any two elements.

For example, suppose I have one 3 tuple, and just suppose I have two vectors-- x1u1 plus x2u2 plus x3u3, and y1u1 plus y2u2 plus y3u3. Let's suppose I want to find that dot product. Notice that my distributive rule for multiplication, coupled with the commutative properties and how I can factor out scale in multiples, tells me that I multiply these two trinomials the same way as I would have had these been algebraic expressions.

In other words, I dot x1u1 with each of these three terms, I dot x2u2 with each of these three terms, and I dot x3u3 with each of these three terms. For example, this will give me x1y1 times u1 dot u1. But if you recall, I said that u1 dot u1 is 3, so this becomes 3x1y1. And in a similar way, when I dot x1u1 with y2u2, I get x1y2, u1 dot u2. But I gave the fact that u1 dot u2 was 4, so this term becomes 4x1y2.

And in a similar way, going through this, I get-- a term will be 5x1y3. Then, dotting x2u2 with each of these three terms, I get three more terms-- 4x2y1 plus 6x2y2 plus 7x2y3. Then I dot x3u3 with each of these three terms. Again, just to do this thing quickly.

x3u3 dotted with y1u1 is x3y1 times u3 dotted with u1. And u3 dotted with u1 was given to be 5. In fact, it was the same as u1 dotted with u3 by commutativity, you see, which was also 5. At any rate, I get what? 5x3y1 plus 7x3y2 plus 9x3y3.

By the way, at this particular stage, I would like to pause for a moment. And some of you may have picked this up better than others, but at this stage of the game, maybe you've already learned how to look at a system like this and write it in matrix form. Now, notice that these are what we call quadratic forms. Each term involves an x multiplied by a y.

I've written this so that the x's are always the left side factor and the y's are always on the right. Notice that there is a matrix of coefficients-- namely 3, 4, 5, 4, 6, 7, 5, 7, 9. And that consequently, I can write this array of nine terms simply as the following product.

I write the row matrix, x1 x2 x3. Multiply that by the 3 by 3 matrix-- 3 4 5, 4 6 7, 5 7 9, and multiply that, in turn, by the column matrix, y1 y2 y3. As a quick check, notice that this is a 1 by 3 matrix. This is 3 by 3. This is 3 by 1. And consequently, this will be a 1 by 1 matrix, or a number.

What number will it be? It will be this number here. And notice what I'm saying-- that x1, x2, x3, y1, y2, y3, are given numbers. And consequently, the dot product in question is determined as soon as I know these coefficients, which happen to be the dot products of the basis elements with one another.

So this shows two things. One, how we write this in matrix form. And two, how the dot product of each two vectors in v is determined by just knowing what happens to the pairs of basis vectors when we dot them. And this gives us still another way of seeing how matrices are used as a coding system.

In other words, every symmetric-- remember what symmetric means? You get the same matrix when you interchange rows and columns. Every symmetric n by n matrix codes a dot product. And conversely, every dot product is coded by a symmetric n by n matrix. You see, what does this matrix tell me?

This is the entry in the i-- the entry in the i-th row, j-th column is what happens when u sub i is dotted with u sub j. Without meaning to belabor this point, let us notice that the array here-- 3 4 5, 4 6 7, 5 7 9-- is precisely the array that we had over here-- 3 4 5, 4 6 7, 5 7 9. This is what we mean when we talk about the matrix of the dot product. It's the array of how the basis vectors look when you dot them with one another.

By the way, as another aside, we agreed that when a vector with itself, that can be identified in a way with the length of the vector. In particular, going back to this example here, if the x's and the y's are chosen to be equal-- if I let x1 equal y1, x2 equal y2, and x3 equals y3, this problem here reduces to this problem. Which in turn, just combining what I had above-- in other words, with the x's and the y's equal-- gives me that to find the square of the length of a vector, I have to solve this particular algebraic equation.

And this algebraic equation is called a quadratic form. You see, notice that each of the variables, x1, x2, x3, appears in each term here as a second-degree variable. See, here's an x1 squared. Here's x1 times x2. You see two factors-- x1 times x3.

And what I'm driving at, again, is aside from any other connections, notice that to solve for the length of a particular vector, one has to wind up solving what we call quadratic forms. And these, for two and three dimensions, can be identified with certain curves in two- or three-dimensional space. But I don't want to belabor that point too much here, either. I just want you to get a buckshot overview of how one works abstractly with the dot product idea.

By the way, some new terminology. If v is an n dimensional vector space with basis u1 up to u n, we call the basis u1 up to u n an orthogonal basis relative to a given dot product if the dot product of any two different members of the basis is 0. In other words, if u sub i dot u sub j is 0 for all i unequal to j, we call this an orthogonal basis, for the usual reason of what orthogonal means. When the dot product of two arrows was zero, we said the arrows were perpendicular, et cetera. This is just a carryover from that.

In particular, if in addition, the dot product of any vector with itself happens to be 1, then the basis is called an orthonormal basis. And this whole idea is modeled after the usual example that in three-dimensional space, i j k forms an orthonormal basis, relative to the usual dot product. In other words, i dot j, i dot k, and j dot k are all 0. But i dot i, j dot j, and k dot k are all 1.

Now, the interesting point is that in most textbooks, one always defines the dot product in much the same way as we do i, j, and k. One always assumes that somehow or other, we have an orthonormal basis. Now, obviously in the example I'm motivating today's lecture on, we don't have an orthonormal basis. Remember, we had something like u1 dot u2 was 4.

Well, see, to be orthonormal, u1 dot u2, first of all, would have to be 0, and u1 dot u1 would have to be 1. But in our example, u1 dot u1 is 3. So the question that comes up is, do we always have an orthonormal basis? And the answer is yes.

And in fact, if you like big words, the process by which we construct an orthonormal basis-- in fact, the process is easier than the name. The name is called the Gram-Schmidt orthogonalization process. And geometrically, what the method boils down to is that if I'm given two non-parallel vectors, u1 and u2, to find the space spanned by u1 and u2, I can replace u2 by the component of u2, which is perpendicular to u1. Let's call that component u2 star.

What I'm saying is that u1 and u2 star span the same space as u1 and u2. But obviously, geometrically, it's easy to see that u1 and u2 star are orthogonal. Now, the question is, how could we get the same result without having to rely on geometry? And the answer is, we say, look it.

We know already in our course that in terms of spanning vectors, if you replace a vector by itself plus or minus a scalar multiple of another, you don't change the space spanned by the given set of vectors. So why not write u2 star to be u2 minus some suitable multiple of u1? See, u2 minus xu1, where we'll try to determine x in such a way that u2 star, dotted with u1, will be 0.

You see, what we're going to try to do is replace u2 by an equivalent vector, meaning that u1 and u2 star will span the same space as u1 and u2, but with the additional property that u1 and u2 star will be orthogonal. Whereas, u1 and u2 might not have been orthogonal. And now you see, we can proceed axiomatically.

Namely, we say, OK. What we want to look at is u2 star dot u1. So let's start both sides of this equation with u1. You see, the left-hand side will become u2 star dot u1. The right-hand side becomes u2 minus xu1, 1 dotted with u1. But by our distributive property, this is the same as u2 dot u1 minus xu1 dot u1.

Now, we didn't want x to be any old number. We wanted x to be that number, if one exists, that makes u2 star dot u1 equal to 0. Remember, u2 dot u1 and u1 dot u1 are numbers. Consequently, I treat this as an ordinary algebraic equation. Namely, this is a number. This is a number. I equate this to 0 and solve for x. In other words, x is u2 dot u1 divided by u1 dot u1.

And by the way, just as an aside, many mathematicians, as an abbreviation when they want to write the dot product of a vector and itself, they write it as the square of the given vector. Obviously, this cannot mean the usual squaring operation, because we don't multiply vectors by themselves. This obviously refers to the dot product. And I mention this in passing, simply so that if you see this in the literature, you will not be confused by it. I prefer to write u1 dot u1 rather than u1 squared, but we'd like you to see this notation anyway.

At any rate, now knowing what x is, I replace x here into this equation, and I wind up with that u2 star is u2 minus u2 dot u1 over u1 dot u1 times u1. And if you want to check this thing geometrically, the u2 star that I've computed this way is precisely the u2 star that I get geometrically. The beauty of this technique is that it did not require geometry. And consequently, structurally, I should be able to use this in higher dimensions. And that's exactly what the Gram-Schmidt orthogonalization process is.

What you do is is you start with a basis. You take the first vector in that basis and dot it with the second vector. If that dot product is 0, those two vectors are both eligible to be part of an orthogonal basis. Now, if that dot product isn't 0, you use the Gram-Schmidt orthogonalization process to replace the second vector, u2, by u2 star, where u1 dot u2 star is 0.

Now you take that new basis, which spans the same space as the original u1 and u2, and compare that or take that in connection with u3 and see if those three vectors form an orthogonal basis-- or orthogonal, meaning their dot product of different ones is 0. And if it is, fine. And if it isn't, you just inductively keep using the Gram-Schmidt orthogonalization process.

Now, to show you what this means in a specific case, let's suppose I have the following situation. Let's suppose I have a four-dimensional vector space for which a basis is u1, u2, u3, and u4. Suppose I also know that u1, u2, and u3 form an orthogonal set.

That means I know that u1 dot u2, u1 dot u3, and u2 dot u3 are all 0. But I don't know that u4 dot u1, u4 dot u2, u4 dot u3 are 0 or anything like this. And the question that comes up now is, I would like to replace u4 by an equivalent vector, which I'll call u sub 4 star, such that u1, u2, u3, and u sub 4 star span the same space, v. But that now with u4 replaced by u4 star, this becomes an orthogonal basis.

Well, the way I proceed is I use the fact that whenever I replace a vector by itself, plus or minus a linear combination of the other vectors, I don't change the space spanned by the vectors. So what I say is, let u4 star be u4 minus x1u1 minus x2u2 minus x3u3. You see, I'm modeling this after what I did in the two-dimensional case. What I want to do is determine x1, x2, and x3 such that u4 star dot u1, u4 star dot u2, and u4 star dot u3 will all be 0. And the trick is simply this.

Let me focus my attention on one of these coefficients, because the technique is the same for all the others. Let's suppose I want to find out what x3 is. The trick is, I dot both sides of this equation with u3. Now, why do I do that? Remember, u1, u2, and u3 are an orthogonal set. Consequently, when I dot u1 with u3 and u2 with u3, those terms will be 0, and they will drop out.

In other words, dotting both sides of this equation with u3 and using the fact that the dot product is distributive, I get u4 star dot u3 is u4 dot u3 minus x1u1 dot u3 minus x2u2 dot u3 minus x3u3 dot u3. And the key point is that by orthogonality, these two things here are 0. Consequently, u4 star dot u3 is u4 dot u3 minus x3u3 dot u3. Remember, these are numbers.

I would like u4 star dot u3 to be 0, so all I have to do now is solve this algebraic equation for x3. And notice that x3 algebraically simply turns out to be u4 dot u3 over u3 dot u3. Again, the only thing I have to be careful about is that the denominator not be 0. And notice that by positive definiteness, u3 dot u3 can only equal 0 if u3 is the 0 vector. But if u3 had been the 0 vector, it couldn't be part of a basis.

So you get the idea of how this works. This is the same Gram-Schmidt orthogonalization process that we used in two dimensions. Because once we know that all the vectors up to u sub 4 are orthogonal, whenever we dot one of them with the other three, we essentially reduce this to a two-dimensional problem, because all the other dot products drop out.

In a similar way, you see, I could have found x sub 2 and x sub 1, leaving the details to you. The easiest way to remember this is every place I see a subscript 3 over here, let me replace it by a 2. And then every place I see the 3, let me replace it by a 1. And what I get is that u4 star is u4. And then I subtract off these scalar multiples.

What scalar multiples are they? Notice that it's what? The coefficient of u1 is u4 dot u1 over u1 dot u1. The coefficient of u2 has as its denominator u2 dot u2. The numerator is u4 dotted with u2. In other words, in a sense, it's like the projection of u4 onto u2. And the coefficient of u3 has as its denominator u3 dot u3, and its denominator is u4 dot u3.

Now, to show you how this works, let's come back to our original problem, which I've reproduced over here. Remember what we had. We had that u1 dot u1 is 3, u1 dot u2 is 4, u1 dot u3 is 5, et cetera. I'll refer to this as I need it. Let me show you how the Gram-Schmidt process allows me to find, from this, an orthogonal basis.

The first thing is that obviously, u1 is not the 0 vector, because if it were the 0 vector, u1 dot u1 would be 0, not 3. So the first vector in my orthogonal basis, which I'll call u1 star, can be u1 itself. How do I find the second vector?

According to the Gram-Schmidt orthogonalization process, I replace u2 by u2 minus a suitable scalar multiple of u1 star. And what scalar multiple is that? The denominator will be u1 star dot u1 star, and the numerator will be u2 dot u1 star. Well, remember, u1 star is equal to u1. So right from this array here, what do I have?

u2 dot u1 star is u2 dot u1. u2 dot u1 is 4. u1 star dot u1 star is u1 dot u1. u1 dot u1 is 3. So this is simply what? Minus 4 over 3 times u1. In other words, u2 star is simply u2 minus 4/3 u1.

And by the way, as a check, what was this supposed to do? If I replaced u2 by u2 star, these two vectors here should now be orthogonal. Are they? Let's start them. If I dot these two, I get what? u1 dot u2 minus 4/3 u1 dot u1.

Notice that u1 dot u2 is 4. u1 dot u1 is 3. So coming down here as a check, u1 star dot u2 star is u1 minus u2, which is 4, minus 4/3 u1 dot u1, which is 3. The 3's cancel. 4 minus 4 is 0. This checks out fine.

How do I find u3 star? Well, I subtract from u3 suitable scalar multiples of u1 star and u2 star. You see what I've done. I orthogonalized one vector at a time. See, first I got u2 star. Now I know that u1 star and u2 star are orthogonal, using u1 star and u2 star, the multiples that I subtract from u3 are what?

For u1 star, my denominator is u1 star dot u1 star, and my numerator is u3 dot u1 star. And for u2 star, my denominator is u2 star dot u2 star, and my numerator is u3 dot u2 star. And now how do I evaluate what this thing really means?

Well, first of all, I know what u3 dot u1 star is. That's just u3 dot u1, which happens to be 5. And I also know what u1 star dot u1 star is. That's u1 dot u1, which is 3. I now have to compute what? u3 dot u2 star, and I have to compute u2 dot u2 star.

Keep in mind that I've already computed that u2 star is u2 minus 4/3 u1. Consequently, u3 dot u2 star is u3 dot u2 minus 4/3 u1. And now you see the mechanical algebra takes over again. I know what my axioms for dot products are.

This gives me u3 dot u2, minus 4/3 u3 dot u1. I'm given that u3 dot u2 is 7. I'm given that u3 dot u1 is 5. So u3 dot u2 star is 7 minus 4/3 of 5. That's 1/3-- 21 minus 20 over 3, which is 1/3.

So let's see what I know now. I know now how to handle this numerator. u3 dot u2 star is 1/3. How do I handle u2 star dot u2 star? u2 star is u2 minus 4/3 u1. So to find u2 star dot u2 star, I have to dot u2 minus 4/3 u1 with itself.

If I do that-- and again, notice how these rules work the same way as for ordinary algebra. I get u2 dot u2 minus 8/3 u1 dot u2 plus 16/9 u1 dot u1. Well, I know that u2 dot u2 is 6. That was given. I know that u2 dot u2 is 6, u1 dot u2 is 4, and u1 dot u1 is 3.

So evaluating this, I wind up with the fact that u2 star dot u2 star is 6 minus 16/3, or 2/3. Consequently, going back here, you see my numerator is 1/3. My denominator is 2/3, so the fraction is 1/2. In other words, substituting into here, I get that u3 star is u3 minus 5/3 u1 minus 1/2 u2 minus 4/3 u1.

And retabulating my results to write them in the usual order of u1, u2, and u3 components, notice that u1 star is u1. u2 star is minus 4/3 u1 plus u2. u3 star is minus u1 minus 1/2 u2 plus u3. Now, what property does this set of vectors have?

First of all, my claim is that they're orthogonal. And secondly, they span the same space as u1, u2, and u3. And it's clear why they span the same space as u1, u2, and u3. Namely, they're essentially suitable scalar multiples of u1, u2-- linear combinations of scalar multiples of u1, u2, and u3.

As a quick check-- and I'll leave some of the other details to you. Let's actually check that u1 star dot u3 star is really 0. If I dot u1 star with u3 star, look at what I get. I get minus u1 dot u1 minus 1/2 u1 dot u2 plus u1 dot u3.

u1 dot u1 is 3. u1 dot u2 is 4. So this is minus 3 minus 1/2 of 4. That's minus 2. u1 dot u3 is 5, minus 3 minus 2 plus 5 is 0. A similar check will show that u2 star dot u3 star is 0.

We should also check to see what the lengths of u1 star, u2 star, and u3 star are. In other words, u1 star dot u1 star is what? That's u1 dot u1, which is 3. u2 star dot u2 star-- well, we just found that over here. That's 2/3. u3 star dot u3 star-- well, u3 star was minus u1 minus 1/2 u2 plus u3.

Dotting that with itself-- and notice I used the square notation here to save space and not have this run too long. But dotting this, the same as I would the square of any trinomial, I get what? u1 dot u1 plus 1/4 u2 dot u2 plus u3 dot u3 plus u1 dot u2 minus twice u1 dot u3 minus u2 dot u3.

Putting in the values for u1 dot u1, for u2 dot u2, u3 dot u3, u1 dot u2, u1 dot u3, and u2 dot u3-- putting in these values, I find that u3 star dot u3 star is 1/2. So I have an orthogonal basis. It's not orthonormal, but notice that it's easy to fix up the normality part, because all I have to do is divide each of these vectors by the square root of this number, and that will make the dot product 1.

For example, if I were to replace u1 star by u1 star over the square root of 3, then when I dotted these two vectors, I would have 3 divided by the square root of 3 times the square root of 3, which is 3. And 3 over 3 is 1. So getting the lengths to be one is quite easy. The hard part is this Gram-Schmidt orthogonalization process to orthogonalize what's going on.

Now, this may seem rather complicated, but it's really one long piece of computation. We'll do this more slowly in the exercises. But the whole idea is what? I successively use the Gram-Schmidt orthogonalization process to replace each vector by a suitable linear combination of itself and the preceding ones to make sure that the new replacement is orthogonal to the remaining ones, and that doesn't change the space that I already have.

By the way, if you want to see what this means in terms of matrix multiplication, notice that we saw earlier that to dot a vector with itself, you write the vector as a row vector, then write down the matrix of coefficients for the basis elements, then write down that same vector as a column vector. The fact that u1 star is u1 plus 0 u2 plus 0 u3 means that as a 3 tuple, it would be 1 0 0. It's transpose would be 1 0 0-- the column vector.

And so when we take this vector and dot it with this vector, that should be the entry, u1 dot u1, in this particular matrix here. If we take this times this, this should be u1 star dotted with u2 star. And since u1 star and u2 star are orthogonal, that had better come out to be 0.

And going on in this way, what I claim is, if you now replace u1 star, u2 star, and u3 star by what 3 tuples they are, then also replace them as column vectors here, you will get a new 3 by 3 matrix which will be diagonal. In other words, if we change our basis from u1, u2, and u3-- if we change that basis to u1 star, u2 star, u3 star, notice that the new matrix that we get is a diagonal matrix. It's our way of saying that when you dot two different vectors, the dot product is 0.

The diagonal elements tell you what u1 star dot u1 star are-- u2 star dot u2 star, u3 star dot u3 star. You see, by using the diagonal matrix over here, this is a much simpler model to use. u1 star, u2 star, u3 star is a much nicer basis to use than u1, u2, and u3 in order to determine what dot products are. I'll emphasize that in more detail later.

Let me make one remark before I get to our concluding remarks. And that is, have you begun to notice how complicated matrices are from the point of view that they are such a wonderful coding device that we use them to code many different things? For example, we have used matrices to code a linear transformation relative to various bases.

We're now using matrices to code the same dot product relative to different bases. We have used matrices to code equivalent row-reduced bases as bases for the same vector space. That at the end of this unit, what I will do is try to give you a summary in the study guide of the different kinds of matrix equivalents and where they're used. Because after a while, you tend to get mixed up if we don't see these pieces one bit at a time.

But to help you see what all this problem means, I have cheated-- that the place I got the data from for the u1, u2, and u3 in this problem was that I thought of the following three vectors-- i plus j plus k, 2i plus j plus k, 2i plus j plus 2k-- and used this to make up the data for u1, u2, and u3 that was in this exercise. What you may find informative, and what I will do as a learning exercise, is have you go through the same exercise that we have just done as today's lesson and have you verify that this checks out with the equivalent geometric construction-- that one of the exercises in the homework will be to show what the Gram-Schmidt orthogonalization process means geometrically for these three vectors.

By the way, rather than to belabor that point for now, let me make one other aside. And that is, I had mentioned before that to find the dot product of two vectors, it may be more advantageous to use one basis rather than another. Remember, when we wanted to dot a vector with itself, using u1, u2, and u3 as a basis, remember we got six different terms? We had an x1 squared, x2 squared, x3 squared, an x1x2, x1x3, and x2x3 term.

Notice that when you pick an orthogonal basis, you only get three terms. Because once the basis is orthogonal, all you have to do is dot corresponding entries. Because you see, if I dot u1 star with a term involving u2 star or u3 star, that will be 0 by orthogonality. If I didn't have an orthogonal basis, I couldn't neglect those terms.

So for example, to dot a vector with itself in this particular example, using u1 star, u2 star, and u3 star as a basis, notice that what I wind up with is what? x1 times x1. x1 squared times u1 star dot u1 star, which is 3, et cetera. Meaning 3x1 squared plus 2/3 x2 squared plus 1/2 x3 squared, which is the square of the length of the vector that I'm looking for.

If I put everything over a common denominator, I get a very nice equation for an ellipsoid over here. You see a very nice equation with no mixed terms-- just perfect squares appearing over here. The mixed terms drop out, and that's one reason why we like orthogonal bases.

Still another reason that we like orthogonal bases-- and I'll conclude with this example-- is that if we know that the vectors u1 up to u n are orthonormal, say, and that x1u1 plus et cetera x n u n is 0, then we automatically know that the u's are-- that x1 up to x n are 0-- that these are linearly independent. And the proof is quite simple.

Namely, all we do is we dot both sides of this equation with, for example, u sub 1. We could've used any one of the u's that we wanted, but using u sub 1, we dot u sub 1 with both sides of this equation. Notice that this term gives me x1u1 dot u1. The next terms are what? u1 dot u2 up to u1 dot u n.

By the property of being orthogonal, all of these products are 0. Consequently, this scalar times x1 must be 0. Consequently, x1 itself must be 0. I've deliberately gone very rapidly here to give you an overview, and to also have you see, as I'm talking, what these computations look like.

I think by hearing this, as complicated as it may sound, I think when you now read the solutions to the exercises, you will have a better feeling for what's going on. And as I say, in the exercises, I will develop these topics much more slowly, and you can read these solutions as we go along. At any rate, this presents our overview of dot products. And so until next time, then, goodbye.

Funding for the publication of this video was provided by the Gabriella and Paul Rosenbaum Foundation. Help OCW continue to provide free and open access to MIT courses by making a donation at ocw.mit.edu/donate.