HERBERT GROSS: Hi. We've sort of arrive at D-day in our course; that in a manner of speaking, everything that we've done up to now has been a rehearsal. That today we are going to come to grips with what the course is all about: namely, a real valued function of several real variables. And before getting into that, what I'd like to do is to review very briefly what we've talked about so far in terms of functions, since we've introduced the vector notation.

Namely, what we've mentioned is, is that our function machine could have either a vector, or scalar as its input, and either a vector, or a scalar as its output. Part one of our course centered around the idea where both the input and the outputs were scalars. Whereas, the previous block of material concerned the case where our input was a scalar, and that output was a vector. Today what we're going to discuss is the situation that occurs when our input is a vector, and our output is a scalar.

And I call today's lecture "N-dimensional Vector Spaces." Eventually, we'll discuss vector spaces in more detail. For the time being, I simply want to set the mood, and hopefully by the time I'm through, show you a rather peaceful coexistence between the worlds of the new mathematics, and the worlds off traditional mathematics.

In fact, in many of our topics that we're going to tackle in this block of material, we will give both points of view. But to set the stage properly-- to get into the idea of what a vector space is all about-- and if that would frightens you just don't worry about for a minute or two. Worry about it after that, but let's just get started.

Let's consider the situation where I have a function machine where my input is a vector and my output is a scalar. As an example, let me just generically let v represent the vector whose form is xi plus yj, using i and j components. Let me

define f(v) to be pi times the square of the i component times the j component.

And let's not worry right now about why I picked this particular recipe. Let's simply observe that once this recipe is chosen, f is a function which maps a two dimensional vector here in i and j components, into a number of pi x^2 y. Just to illustrate this recipe, notice that if our vector were 3i plus 4j, the output of the f machine-- if this were the input-- would be what?

Pi times the square of the first component-- the component of i, that's 3 squared-- times the component of j, which is 4. And that leads to 36 pi. By the way, observe that order does make a difference, namely if I reverse the roles of the coefficients, and feed the vector 4i plus 3j into my f machine, the output would be pi times 4 squared-- the square of the i component-- times 3, and that would be 48 pi.

Notice also that we have allowed already in our course the abbreviation that (a, b) would represent ai plus bj, or (a, b, c) would represent ai plus bj plus ck. The idea is that if we now apply this shorthand notation to these two vectors, what we could say is what? f of... this vector, f(3,4) is 36 pi, whereas f(4,3) is 48 pi. And that's exactly what we mean when we say that we may treat a two dimensional vector as an ordered pair. You see, it's, not only a pair but the order does make a difference. Both in what the vector is and what the output of the f machine is. OK?

Hopefully, let's say, so far so good, and let's tackle now a rather completely different problem. What I'd like to do now is the following-- let's consider the cylinder-- the right circular cylinder-- the radius of whose base is x, and whose height is y. Notice that the volume of this cylinder is pi x^2 y.

And if I use the same f that we used previously-- in other words, the same f that we were talking about over here. Notice that another way-- see how is f defined? Given that the input was (x, y), the output was pi x^2 y, notice that the volume is f(x,y). In particular, if I want the volume of the cylinder, the radius of whose base is 3, and whose height is 4, see x is 3 and y is 4.

What I really want is f(3,4). That's pi times 3^2 times 4. f(3,4) is 36 pi. Now I'd like to

pause for a second again, and return to our earlier remark. Namely, if I look at this, and if I look at this.

Notice that these two expressions are identical. I cannot tell the difference between whether I'm looking at the vector 3 i plus 4 j, or whether I'm looking at the cylinder, the radius of whose base is 3, and whose height is 4, whether I look at this equation, or whether I look at this equation.

The difference is that in the first case somehow or other, it was quite natural to think of (3, 4) as being either an ordered pair, or an arrow. In this case, however, my contention is that when we think of the radius of the base of the cylinder, and the height, we do not tend to think in terms of arrows, but rather in terms of ordered pairs.

In other words, the ordered pair (x, y) in the expression f(x,y) need not be viewed as an arrow, but as an ordered pair. And an ordered pair is called a 2-tuple. This leads to a generalization that I think is rather important, and I think you will see in a moment, where the idea of this approach comes into functions of several real variables.

The topic I have in mind is something called an n-tuple. And let me read into that rather gradually as follows. Without giving you a specific physical example-- meaning I'll give you an illustration, but leave the numerical amounts out. Quite possibly if I'm studying temperature in a room, the temperature will in general what? It will depend on what position I'm in the room, and also at what time I measure the temperature.

It's fair to assume that in many applications the temperature is some function of the four independent variables x, y, z, and t, where x, y, and z are the Cartesian coordinates of three dimensional space, and t represents time.

What I'm driving at is I can now visualize this in terms of my function machine again. Namely to compute t, I think of feeding what? Specific values into the machine for x, y, z, and t. The f machine then performs on x, y, z, and t as indicated by f to

compute t.

The input of my f machine in this case is what I'm going to call a 4-tuple for the time being. I need four values-- x, y, z, and t. Order does make a difference. For example, if I interchange the x and the y-coordinate, those x and y, what I'm doing is I'm interchanging the x and the y-coordinate of the point in space, and that in general is going to change the point in space.

The point, however, is that in this particular f machine, notice that my output is a scalar. Namely the temperature is a number, but the input is a 4-tuple. x, y, z, and t. Now the trouble with using symbolism like x, y, z, and t-- I guess without going into a long philosophic discussion-- among other things as soon as you have 27 or more independent variables you run out of letters of the alphabet.

As a result, it is quite common for one to adopt a new notation. instead of saying let x, y, z, t be a 4-tuple, what one usually does is chooses one symbol-- say x-- and then uses subscripts. a general 4-tuple would have the form what? (x_1, x_2, x_3, x_4), where x_1, x_2, x_3 and x_4 are numbers.

An expression like this is called the 4-tuple. What is this a generalization of? The 4-tuple is a generalization of the one dimensional, two dimensional, and three dimensional arrow, where we could think of what? The vector x_1 i as just needing one number to specify it. The vector x_1 i plus x_2 j could've been used to do what?

It could've been abbreviated by the 2-tuple (x_1, x_2), and the vector x_1 i plus x_2 j plus x_3 k could've been abbreviated by the 3-tuple (x_1, x_2, x_3). It is conventional in one, two, or three dimensional space to use x, y, and z, instead of x_1, x_2, and x_3. But that's just a convention. I think that it's because we learnt it that way that we do it. In general, I think the subscript notation is much nicer, but in general the idea is what? Given an ordered array of n numbers, x_1 up to x_n, we call that an n-tuple.

And my friend and colleague John Fitch mentioned to me that if n is odd, like one, three, five, or seven, then it's known as an odd-tuple. Which isn't a very funny story, that's why I told you John told me that particular story. But the whole idea is this is

an n-tuple. And the whole idea again is what? That an n-tuple makes sense, even when n is greater than three.

The whole name of the game of functions of several real variables-- in terms of modern mathematics, in terms of the language of n-tuples-- is that a real valued function of several, where by several you mean more than one, real variables is simply a function in which the input is an n-tuple and the output is a number. OK?

That's what this whole thing is all about. And because of that, when we then abbreviate the n-tuple, we use x with a bar under it. Let's call it x-bar. Rather than x with the arrow over it, since arrows may be inappropriate. Now what do I mean by inappropriate? Well I mean that even in the case of one, two, or three dimensions, you might be thinking of say the radius, and the height of a cylinder, rather than as an arrow.

And in more than three dimensions-- for most of us at least-- it's difficult to visualize what we would mean by an arrow. So we just use the bar underneath. Now again, the major point is notice this-- I keep saying the major point. I guess there's a lot of major points about this.

Remember that we did not call arrows "vectors". We did not call arrows "vectors" until we defined a structure on the arrows. Remember what we did? We told what it meant for two arrows to be equal, we told how we added two arrows, and we told how we multiplied an arrow by a scalar.

In a similar way, we will not call n-tuples a structure until we tell how to equate a pair of n-tuples, how to add a pair, and how to multiply an n-tuple by a number. By the way, the structure that we wind up with is then called an n-dimensional vector space, or more concisely, n-space.

And the idea works like this-- let's pick a particular value of n. Lets just call it n. And let s sub n be the set of all n-tuples x1 up to xn, a set of what? All n-tuples of numbers x1 up to xn. Let's pick two particular members of s sub n, which we'll call a-bar and b-bar. Where a-bar is simply an abbreviation for a, the n-tuple a1 up to

an; where the a1, a2 up to an, et cetera are real numbers. And b-bar is an abbreviation for the n-tuple b1. et cetera, bn; where b1 up through bn are also real numbers.

Now again, here's where structure comes into play. We have already defined an n-tuple arithmetic in terms of arrows for the case when n is either one, two, or three.

Based on what happens when n is one, two, or three, we invent the following definitions. First of all, we invent the definition that a-bar equals b-bar means that the components-- meaning what? The individual members of the n-tuple-- the components of a-bar are equal to the components of b-bar, component by component.

In other words, a1 is equal to b1, a2 is equal to b2, et cetera. All the way up to what? an is equal to bn. Now, in other words again, what we're saying is that for two n-tuples to be equal, by definition, they should be equal component by component, and this is motivated by the fact that we already know that we've accepted the structural definition for the case of arrows.

Similarly, given two n-tuples a-bar and b-bar, to add them, let me define that to be the n-tuple that I get by adding component by component. In other words, to find the first component of a-bar plus b-bar, I add the first component of a-bar to the first component of b-bar. Noticing of course, that this is what? Be careful here. This is one number.

(a1 + b1) is one number. (a2 + b2) is a number. (an + bn) is a number. In other words notice that by this definition, the sum of two n-tuples is again an n-tuple.

And finally, to multiply a scalar by an n-tuple, I will agree to define that definition to mean that you multiply the n-tuple component by component by that particular scalar, or number. Notice again that all I have done here is I have obtained these three structural definitions from the equivalent situations of arrows.

And since everything that was true about arrows followed from these three basic definitions, any set up n-tuples that obeys this particular structure will also behave

like the arrows did. And that's why we call it a vector space. They behave like vectors even though they can no longer be viewed as arrows. And again there are creative people who view these things as arrows.

I remember feeling very intimidated one day by my undergraduate professor the first time I learned vector spaces. I said, how do you visualize an n-dimensional vector space? And in full seriousness, without batting an eyelash, he says, "I visualize it like a porcupine with a bunch of quills coming out of it." And I knew that he knew what was visualizing it like, but didn't help me one bit.

I'm saying, if you can visualize this things as arrows, be my guest. Feel free to do so. If you can't, notice that every one of these definitions stands on its own two feet. Subject to the condition that when n is one, two, or three, we happen to have a very nice geometric interpretation. By the way, I may have given you the impression that vector spaces were invented because of functions of several variables.

Rather, the impression I would like to leave you with is, that in terms of motivating vector spaces, in terms of this course, that was the motivation that we elected to use. That the mathematician talked about vector spaces in many a different context from what we might even dream possible.

In other words, I don't even have to think of temperature being a function of the four variables x, y, z, and t. Let me give you a different kind of non-trivial example of a four space that doesn't even bring functions into play. Let's suppose I invent the abbreviation, I write the 4-tuple (a_0, a_1, a_2, a_3), to denote the cubic polynomial $a_0 + a_1 x + a_2 x^2 + a_3 x^3$.

Notice that I can use these as a place value system. The first member tells me my constant term, the second member tells me the coefficient of x, the third member tells me the coefficient of x^2, and the fourth number gives me the coefficient of x^3. Notice that for two polynomials to be identically equal, they must be equal, what? Coefficient by coefficient.

That means what? Component by component. How do we add two polynomials?

We add them coefficient by coefficient. We add like terms. In other words, given two polynomials, we add them what? Component by component. We add the two constant terms together, the two coefficients of x together, the two coefficients of x^2 together, the two coefficients of x^3 together. You see?

How do we multiply a polynomial by a scalar? We multiply each term by the scalar. That in turn is equivalent to multiplying each coefficient by that scalar. And that says in terms of n-tuple notation, that we have multiplied each component by that scalar.

The set of polynomials of degree n forms a very nice vector space in terms of our definition of a vector space. Now of course the danger is that one gets the idea that any set of n-tuples can be viewed as a vector space. An n-dimensional vector space. But this we have to be careful about. Remember, it is not the n-tuples, it is structure that they obey.

Let me give you sort of a simple example over here. Let me consider the following situation. First of all, let me just emphasize a statement I just made it, let me just read it with you. n-tuples are not automatically n-spaces. For example, let me invent the 2-tuple (a, b) to represent the number a + b.

For example, if I define the 2-tuple (a, b) to be an abbreviation for a + b, What would (4, 5) denote? Remember the 2-tuple means what? To get the value of the 2-tuple is just the sum of the components. If I add a and b, in this case, 4 + 5 happens to be 9. How about the 2-tuple (6, 3)? What value would that have? That would also have the value 9.

Therefore numerically, the 2-tuple (4, 5) is equal to the 2-tuple (6, 3). Yet notice that the first component is not equal to the first component here. In other words 4 is not equal to 6. Nor is 5 is equal to 3, but if I were to choose this definition of equality, I could not say that these 2-tuples form a two dimensional vector space, because it violates the first definition for a vector space, namely the definition of what it means for two vectors to be equal.

Since we're going to let most of our material be covered by the exercises and the

supplementary notes, and this is just to be an overview, let's move on now. Let's assume that we now know what n-dimensional vector spaces are like. We now know that we can view functions of several variables as functions that map n-tuples into numbers, and as a result it now makes sense to talk about things like suppose you were given the n-tuple (x1, x2, x3, x4), and supposed that under f, that n-tuple was mapped into x1^3 + x2 + x3^2 + 2*x4.

For example, if I were to replace x1 by 1, x2 by 3, x3 by 1, and x4 by 2, I would arrive at the result what? 1^3 + 3 + 1^2 + 2 * 2, and I can compute that output.

Now the question that comes up in calculus is, can we talk about limits here? Instead of computing what f(1,3,1,2) is, can I compute the limit of this thing as (x1, x2, x3, x4) approaches (1, 3, 1, 2)?

I think intuitively it's clear that since equality means that you must have equality component by component, to say that this approaches this means that the first component here must approach the first component here. The second component here approaches is the second component here et cetera.

In other words, this could be replaced by the four separate linear one dimensional limit problems: x1 approaches 1, x2 approaches 3, x3 approaches 1, and x4 approaches 2. And we would then be tempted to say what? We will replace x1 by 1. x2 by 3, x3 by 1, x4 by 2. See what happens to this expression?

And we would then be tempted to say that this particular limit was equal to 9. Now the interesting point is this-- that traditionally, this particular problem was tackled long before anyone invented vector spaces. Or at least long before anybody was serious about vector spaces.

People did say, why can't we reduce the study of four dimensional space to four separate studies of one dimensional space? In other words let x1 approach 1, x2 approach 3, in that case you're allowing what? Four separate one dimensional limits to be taking place here.

But the insight that modern math gave us was that we can now go back to our

9

traditional definition of limit. Remember what was our old structural definition of limit? Way back from the first time we had it. The limit of f(x) as x approaches a equals L means, given epsilon greater than 0, we can find delta greater than zero, such that whenever the absolute value of x - a is greater than 0 but less than delta, the absolute value of f(x) - L is less than epsilon

Now here's what the new math said. The modern approach said look, let's just take our old structural definition-- the same as before-- and vectorize everything. Notice in this situation that we're dealing with the input is where we have the several variables. The input is the n-tuple, the vector OK? And the output is the scalar.

So f is a scalar, L is a scalar. But x and a are vectors, so every place I see an x and an a, I have to put the bar underneath. And now I read this definition, and all of a sudden, as so often has happened in our course up to now, I come to something that I've never seen before. Namely, as soon as I look at this.

This made very good sense when these were arrows. We talked about this earlier in our course and one of our lectures. That to say the two arrows were near each other was to say that their difference was small, and that in turn said if the arrows were placed tail to tail, we could make the distance between their heads as small as we wish.

Now the price that we have to pay for higher dimensions is that if we have more dimensions than what we can draw arrows in, the problem that we're faced is that we have not defined what you mean by the magnitude of x - a where x and a happen to be n-tuples.

And here again, we come back to our structure. But now for the first time the structure is not redundant. Let me tell you what I mean by that.

In the one dimensional case we define the magnitude of x - a to be the square root of x1 - a1^2 where the vector x was the 1-tuple x1, and the vector a was the n-tuple a1. In the two dimensional case, we said, OK, let's define the magnitude of the vector x - a to be $(x1 - a1)^2 + (x2 - a2)^2$.

And in the three dimensional case, we said, let's define the magnitude of x-bar - a-bar to be the square root of (x1 - a1)^2 + (x2 - a2)^2 + (x3 - a3)^2. At that time, I kept saying, notice that these recipes do not depend on a picture, that these are numerical results that we can compute without having to draw a picture at all.

What happened of course was that in the one dimensional case, in the two dimensional case, in the three dimensional case, it was easier to visualize the picture. Now here's where the real kicker comes in-- and this is the real crucial point-- structurally, can't you see what's happening over here? Can't you see how I can now define the absolute value of the vector x-bar - a-bar? Even if n is greater than 3 in such a way that the definition will make sense and still mimic everything that we're doing?

I hope you are a step ahead of me on this except for some new notation I introduced here. It turns out that in the modern math book, one distinguishes between the absolute value of a number, and the magnitude of a vector, and it is frequently traditional to introduce a double bar on each side to represent the magnitude of the difference between two vectors,

which I claim behaves like a distance. Let me show you what I mean by that. Let me define the magnitude of the n-tuple x-bar minus the n-tuple a-bar, written this way, to be the positive square root of (x1 - a1)^2 plus et cetera plus (xn - an)^2.

The thing that I would like you to notice here is that since each of these numbers are non-negative, see they're squares of real numbers, the only way this can be 0-- well the only way that the sum of squares of non-negative numbers can be 0 is for each of the numbers to be 0. Consequently, the only way the magnitude of x-bar - a-bar can equal 0 is if x1 equals a1, x2 equals a2, et cetera. And xn equals an.

In this vein, notice that the geometric phrase x-bar near a-bar still makes sense. It doesn't make sense pictorially, because we can't draw the arrows if n is greater than 3. But notice that what we're saying is that for x to be near a, all we're saying is that the magnitude defined this way-- the magnitude of x-bar minus a-bar is small.

When you're adding up positive squares, the only way the sum can be small is if each of the factors are small. But notice what these factors are, except for the square, it's the difference between x1 and a1. x2 an a2, et cetera, xn and an.

In other words, to say that x-bar is near a-bar means that x1 is near a1, x2 is near a2, et cetera, and xn is near an, which is exactly the traditional approach. And in fact, except for the fact that we can capitalize on structure, notice that once we define the magnitude of the difference between two n-tuples-- do you notice that by the way? The magnitude of the difference of two n-tuples is a number. Notice now if we replace this fancy phrase-- which we didn't know the meaning of before but which we now know-- by it's new definition, we obtain the traditional definition of limit.

Namely the limit of f(x1...xn) as x1 approaches a1 et cetera, and xn approaches an equals L means that given epsilon greater than 0, we can find delta greater than 0 such that whenever the square root of $(x1 - a1)^2$ plus et cetera $(xn - an)^2$ is less than delta but greater than 0, then the magnitude-- you see these numbers here-- the magnitude of f(x1...xn) - L is less than epsilon.

In other words, this definition here happens to be the traditional definition. OK? But the point is that the traditional definition has exactly the same structure as the modern definition. And as a result-- to make fun of the traditional math because it's not as pretty as the modern math-- is the wrong thing to say. It's like the fellow who once asked me at a PTA meeting how much is 8 + 7 in the new mathematics? That part hasn't changed.

The beauty of using the n-tuple notation was that it allowed us to use the previous structure of limits. So that we can get all of our theorems, all of our formulas and what have you, to go through word for word, even though the higher the dimension, the more complex our computations are. But structurally, It essentially boiled down to, after you've seen one dimensional space you've seen them all.

That was the big innovation with the modern approach to n-dimensional vector spaces. And to help put this in proper perspective, next time I shall introduce the

calculus of several real variables in terms of the more traditional approach. But again, until next time, good bye.