VECTOR-VALUED FUNCTIONS

MATH 195, SECTION 59 (VIPUL NAIK)

Corresponding material in the book: Sections 13.1. 13.2.

What students should definitely get: Definition of vector-valued function, relation with parametric description of curves, basic operations on vector-valued functions, limit and continuity definitions and theorems, definition of derivative and integral, notion of tangent vector.

What students should hopefully/eventually get: Top-down and bottom-up descriptions of curves, finding intersections of curves with various kinds of descriptions, particularly in $\mathbb{R}^2$ and $\mathbb{R}^3$.

Executive summary

0.1. Vector-valued functions, limits, and continuity.

(1) *Not for review discussion*: A vector-valued function is a function from $\mathbb{R}$, or a subset of $\mathbb{R}$, to a vector space $\mathbb{R}^n$. It comprises n scalar functions, one for each of the coordinates. For instance, given scalar functions $f_1, f_2, \dots, f_n$, we can construct a vector-valued function $f = \langle f_1, f_2, \dots, f_n \rangle$ defined by $t \mapsto \langle f_1(t), f_2(t), \dots, f_n(t) \rangle$.

(2) *Not for review discussion*: A vector-valued function in n dimensions corresponds to a parametric description of a curve in $\mathbb{R}^n$ whose points are just the heads of the corresponding vectors. The vector-valued function from the previous observation has corresponding curve $\{(f_1(t), f_2(t), \dots, f_n(t) : t \in D\}$ where D is the appropriate domain.

(3) To add two vector-valued functions in n dimensions, we add them coordinate-wise, where the corresponding scalar functions are added pointwise as usual. This sum is also a vector-valued function in n dimensions.

(4) We can multiply a scalar function and a vector-valued function to get a new vector-valued function. At each point in the domain, this is just multiplication of the corresponding scalar number and the corresponding vector.

(5) We can take the dot product of two vector-valued functions in n dimensions. The dot product is a scalar-valued function. At each point in the domain, it is obtained by taking the dot product of the corresponding vector values.

(6) For $n = 3$, we can take the cross product of two vector-valued functions and get a vector-valued function. This cross product is taken pointwise.

(7) To calculate the limit of a vector-valued function at a point, we calculate the limit separately for each coordinate. We use this idea to define the *limit*, *left hand limit*, and *right hand limit* at any point in the domain.

(8) Limit theorems: Limit of sum is sum of limits, constant scalars pull out of limits, limit of scalar-vector product is product of scalar limit and vector limit, limit of dot product is dot product of limits, limit of cross product (case $n = 3$) is cross product of limits.

(9) A vector-valued function is *continuous* at a point in its domain if each coordinate function is continuous, or equivalently, if the limit equals the value. We say it is continuous on its interval if it is continuous at every point in the interior of the interval and has one-sided continuity at one of the endpoints.

(10) Continuity theorems: Sum of continuous vector-valued functions is continuous, product of continuous scalar function and continuous vector-valued function is continuous, dot product of continuous vector-valued functions is continuous, cross product (case $n = 3$) of continuous vector-valued functions is continuous.

(11) There is no n-dimensional analogue of the intermediate value theorem, multiple things fail.

Actions ...

(1) If no domain is specified, the domain of a vector-valued function is the intersection of the domains of all the constituent scalar functions.

0.2. Top-down and bottom-up descriptions.

0.2. Top-down and bottom-up descriptions. Words ...

(1) A top-down description of a subset of $\mathbb{R}^n$ is in terms of a system of equations and inequality constraints. Each equation (equality constraint) is expected to reduce the dimension by 1 (we start from n) whereas inequality constraints usually have no effect on the dimension. So if there are m independent equality constraints describing a subset of $\mathbb{R}^n$, we expect the subset to have dimension $n - m$.
(2) A bottom-up description is a parametric description with possibly more than one parameter. The number of parameters needed is the dimension of the subset. The parametric descriptions we have seen so far are 1-parameter descriptions and hence they describe curves – 1-dimensional subsets.
(3) The codimension of a m-dimensional subset is $n - m$.
(4) When intersecting, codimensions are expected to add. If the total codimension we get after adding is greater than the dimension of the space, the intersection is expected to be empty.
(5) In $\mathbb{R}^3$, curves are one-dimensional, surfaces are two-dimensional. Thus, curves are not expected to intersect each other, but curves and surfaces are expected to intersect at finite collections of points (in general).

Actions ...

(1) Strategy for finding intersection of subsets in $\mathbb{R}^n$ (specifically, curves and surfaces in $\mathbb{R}^3$) given with top-down descriptions: Take all the equations together and solve simultaneously.
(2) Strategy for finding intersection of curve given parametrically and curve or surface given by top-down description: Plug in the functions of the parameter for the coordinates in the top-down description.
(3) Strategy for finding intersection of curves given parametrically: Choose different letters for parameter values, and then equate coordinate by coordinate. We get a bunch of equations in two variables (the two parameter values).
(4) Strategy for finding collision of curves given parametrically: Just equate coordinates, using the same letter for parameter values. Get a bunch of equations all in one variable.

0.3. Differentiation, tangent vectors, integration.

(1) The derivative of a n-dimensional vector-valued function is again a n-dimensional vector-valued function. It can be defined by differentiating each coordinate with respect to the parameter, or by using a difference quotient expression. These definitions are equivalent.
(2) This derivative operation satisfies the sum rule, pulling out constant scalars, and product rules for scalar-vector multiplication, dot product, and cross product (case $n = 3$).
(3) As a free vector, the tangent vector at $t = t_0$ to a parametric description of a curve is just the derivative vector for the corresponding vector-valued function. As a localized vector, it starts off at the corresponding point in $\mathbb{R}^n$.
(4) The tangent vector for a curve with parametric description depends on the choice of parameterization. The *unit tangent vector* does not, apart from the issue of direction (forward or backward). The unit tangent vector is a unit vector (i.e., length 1 vector) in the direction of the tangent vector. It is unique for a given curve (independent of parameterization) up to forward-backward issues.
(5) To perform definite or indefinite integration of a vector-valued function, we perform the integration coordinate-wise.

1. Vector-valued functions, parametric descriptions and more

As is my wont, I will, wherever possible, state things in n-dimensional terms and then discuss any geometric significance of the case where $n = 2$ or $n = 3$. As mentioned in a previous lecture, restricting to $n = 3$ is a somewhat artificial thing to do from the perspective of the social sciences because the number of quantities that we are interested in simultaneously studying is often substantially more than 3.

1.1. **Vector-valued functions and parametric descriptions of curves.** We hinted at this last time, when motivating vectors, but let's make this formal.

A *vector-valued function* in n dimensions on a subset D of $\mathbb{R}$ is a collection of n functions $f_1, f_2, \dots, f_n : D \to \mathbb{R}$, which are pieced together as coordinates of a vector as follows:

$$t \mapsto \langle f_1(t), f_2(t), \dots, f_n(t) \rangle, \qquad t \in D$$

Thus, a vector-valued function is a *vector of functions in the usual sense.*

A vector-valued function corresponds to a parametric description of a curve in $\mathbb{R}^n$, and the curve is simply the set of corresponding points to the vectors:

$$\{(f_1(t), f_2(t), \dots, f_n(t)) : t \in D\}$$

Note that there is the usual distinction between a curve and its parameterization. The curve is simply the subset of $\mathbb{R}^n$, whereas the parameterization is a particular story about how that curve was built. The same curve could admit multiple parameterizations that differ in timing, speed, direction, and choices made at self-intersection points.

1.2. **Domain convention for vector-valued functions.** If we are given a vector-valued function $f = \langle f_1, f_2, \dots, f_n \rangle$ *without* a domain being specified, the domain is implicitly taken to be the *largest possible subset* of $\mathbb{R}$ on which f makes sense. This turns out to be the *intersection* of the domains of the functions $f_1, f_2, \dots, f_n$.

1.3. **The two-dimensional and three-dimensional cases.** We previously examined the case $n = 2$, and this was what we called parametric descriptions of curves in the plane. The case $n = 3$ gives parametric descriptions of curves in space. These are sometimes called *space curves.* We will talk about these a little later, as a follow-up to a general discussion about top-down versus bottom-up approaches to finding subsets in $\mathbb{R}^n$.

1.4. **Multiple inputs and multiple outputs.** There are two ways in which multivariable calculus generalizes single variable calculus. The first is that we can now have *outputs* which are vectors, or tuples of real numbers, instead of single real numbers. The second is that we can have *inputs* which are tuples of real numbers, instead of single real numbers.

Of these, the challenge that we will currently deal with is the outputs challenge. This turns out to be not much of a challenge at all, and the key idea is to simply deal with things one output coordinate at a time.

The other challenge is the inputs challenge, namely, how do we deal with functions *of more than one variable.* This is fundamentally a deeper challenge. One of the ideas is to deal with the function one input at a time, but the other inputs cannot be completely ignored. The upshot of it all is that dealing with multiple inputs is something we will have to defer till a little later in the course.

Okay, now we move to the baby stuff.

2. Operations on vector-valued functions

2.1. **Four kinds of additions.** If f and g are vector-valued functions in n dimensions, given by $f = \langle f_1, f_2, \dots, f_n \rangle$ and $g = \langle g_1, g_2, \dots, g_n$, then $f + g$ is given by the vector-valued function $\langle f_1 + g_1, f_2 + g_2, \dots, f_n + g_n \rangle$. Explicitly, it is given by the function:

$$t \mapsto \langle f_1(t) + g_1(t), f_2(t) + g_2(t), \dots, f_n(t) + g_n(t) \rangle$$

Overall, we have seen four kinds of additions:

- Addition of scalar numbers.
- Addition of scalar-valued functions, which is done *pointwise*, i.e., to add two scalar-valued functions, we add their values at each point in the input domain.
- Addition of vectors, which is done *coordinate-wise.*
- Addition of vector-valued functions, which is done *pointwise and coordinate-wise*, i.e., to add two vector-valued functions, we add the vector values at each point in the input domain. To add these vectors, we in turn do coordinate-wise addition.

2.2. **Scalar-vector multiplication, dot product, and cross product.** Suppose f is a scalar-valued function and g is a n-dimensional vector-valued function. We can define the product fg. This is the function:

$$t \mapsto \langle f(t)g_1(t), f(t)g_2(t), \dots, f(t)g_n(t) \rangle$$

Suppose f and g are n-dimensional vector-valued functions of one variable. The *dot product* $f \cdot g$ is the function that sends each t to the scalar obtained by taking the dot product $f(t) \cdot g(t)$. Thus, $f \cdot g$ is a scalar-valued function of one variable.

In the case $n = 3$, we can define, for 3-dimensional vector-valued function f and g, a vector-valued function $f \times g$. This sends each point t to the vector $f(t) \times g(t)$.

3. Limits and continuity

3.1. **Definition of limit.** Suppose we have a vector-valued function:

$$f(t) := \langle f_1(t), f_2(t), \dots, f_n(t) \rangle$$

Then, for a real number a, we define:

$$\lim_{t \to a} f(t) = \langle \lim_{t \to a} f_1(t), \lim_{t \to a} f_2(t), \dots, \lim_{t \to a} f_n(t) \rangle$$

In other words, we take limits coordinate-wise. We can similarly define various notions such as *left hand limit* and *right hand limit*.

3.2. **Limit theorems.** We have vector limit theorems that are analogous to, and follow from, the limit theorems for scalars. The usual caveats apply with the interpretation of these results.

- The limit of the sum is the sum of the limits.
- The limit of a scalar multiple is the same scalar multiple of the limit.
- The limit of a product of a scalar-valued function and a vector-valued function is the product of the limit of the scalar-valued function and the limit of the vector-valued function.
- The limit of the dot product is the dot product of the limits.
- In the case $n = 3$, the limit of the cross product is the cross product of the limits.

3.3. **Continuous vector-valued functions.** Continuing with the de ja vu, we define a vector-valued function to be continuous at a point if the limit at the point equals the value at the point. Similarly, we define *left continuous* and *right continuous*. We say that a vector-valued function is continuous on an interval if it is continuous at all points in the interior of the interval and has one-sided continuity from the appropriate side at boundary points.

For each of these notions of continuity, a vector-valued function satisfies that notion if and only if all the corresponding scalar functions satisfy that fnotion.

3.4. **Continuity theorems.** We have the following:

- The sum of continuous vector-valued functions is continuous.
- The product of a constant and a continuous vector-valued function is continuous.
- The product of a continuous scalar-valued function and a continuous vector-valued function is continuous.
- The dot product of two continuous vector-valued functions is a continuous scalar-valued function.
- In the case $n = 3$, the cross product of two continuous vector-valued functions is also a continuous vector-valued function.

3.5. **Intermediate value theorem fails to have an analogue.** Recall the intermediate value theorem, which states that any continuous function f that takes a value $f(a)$ at a and $f(b)$ at b must take all values between $f(a)$ and $f(b)$ on inputs between a and b.

This statement fails to have an analogue for vector-valued functions. First, there isn't a very precise notion of *between* for vector-valued functions. In other words, there is no natural total ordering on vectors that preserves the algebraic and geometric properties we are interested in. So it doesn't even make sense to formulate the statement. We *could* come up with some plausible formulations, but they are false.

It *is* true that each of the coordinates passes through all intermediate values. However, it is not necessary that every *vector* of intermediate values is achieved, because the different coordinates could be changing in different ways.

4. Top-down versus bottom-up: revisiting dimensionality arithmetic

4.1. **Top-down in n dimensions.** n-dimensional space, which we sometimes denote $\mathbb{R}^n$, is n-dimensional because there are n degrees of freedom in specifying a point, i.e., we need to provide n real numbers in order to specify a unique point in the space.

There are two ways of constructing subsets of $\mathbb{R}^n$, the *top-down* approach and the *bottom-up* approach.

In the top-down approach, we start with all of $\mathbb{R}^n$. Then, we decide to whittle down. To whittle the subset down, we provide constraints that a point must satisfy in order to lie in the subset. Here are two key things to note:

- As a *general heuristic*, every new constraint that is *a single scalar equality constraint* and is *mostly independent of previously introduced constraints* reduces the dimensionality of the space by 1. So, if we start with 24-dimensional space, and introduce 3 mutually independent scalar equality constraints, then the subset of the whole space that satisfies all of these constraints is expected to be 21-dimensional (whatever that means).
- As a *general heuristic*, a new constraint that is *a single scalar ineuqality constraint* has no effect on the dimensionality. For instance, in 2-dimensional space with coordinates x and y, introducing the constraint $y > 0$ restricts us to the upper half-plane, but this still has full dimensionality, i.e., 2.
- Combining these, we see that as a *general heuristic*, if we have more than n scalar equality constraints in a n-dimensional space, the solution space is expected to be empty – because its dimension is *less than zero*, which isn't possible. The exception arises when there is some hidden consistency or dependency between the equality constraints.

It's worth noting that there are many exceptions. For instance, the single scalar constraint $x^2+y^2+z^2=0$ in three-dimensional space with coordinates (x,y,z) has a solution space that is zero-dimensional (a single point). Similarly, the constraint $x^2+y^2+z^2=-1$ has an empty solution space, while the constraint $(x+y-1)^2+(y+z-1)^2=0$ has a one-dimensional solution space even though $3-1=2$.

However, these are the exceptions that prove the rule, in the following sense. Over the real numbers, setting a sum of squares of a bunch of quantities to be equal to zero is likely secrely compressing the condition that all of them are equal to zero into a single equation. So it's a bit of cheating compression of multiple constraints into what ostensibly looks like a single constraint. This is a sleight of hand.[1]

4.2. **Bottom-up in n dimensions.** In the previous subsection, we discussed an approach where we start with the whole space and then whittle down the set of points under consideration by adding constraints. I dubbed this the *top-down* approach. In constrast, there is a *bottom-up* approach. Here, we start with a clean slate and then draw in new points.

The parametric descriptions of curves, and vector-valued functions, are the *one-dimensional* case of bottom-up descriptions. A parametric description uses *one* real input (one degree of freedom) and traces a curve in $\mathbb{R}^n$ from that.

What analogoue of parametric description can we use to get higher-dimensional subsets of $\mathbb{R}^n$? For instance, how do we obtain parametric descriptions of *surfaces* in $\mathbb{R}^n$, i.e., two-dimensional subsets of $\mathbb{R}^n$?

[1] If you were working over the complex numbers instead of the real numbers, this kind of sleight of hand would not be possible. There are some deep results in mathematics, which you may never see in your life, which basically say that over the complex numbers, some rigorous version of the statements I made above *is* true if we restrict ourselves to things involving polynomials.

The general idea is as follows: a parametric description of a k-dimensional subset involves a k-parameter description. For instance, a parametric description of a surface in $\mathbb{R}^n$ corresponds to a n-dimensional vector-valued function with *two* real inputs, so that each of the coordinates itself is a function of two-variables. Concretely, it looks like:

$$(t, u) \mapsto \langle f_1(t, u), f_2(t, u), \ldots, f_n(t, u) \rangle$$

In other words, we now have *two degrees of freedom for the input.* The input pair is free to vary over some subset of $\mathbb{R}^2$, and the output traces a surface in $\mathbb{R}^n$.

We will return to this idea a little later in the course. It requires a new way of thinking about functions since we now have to deal with functions of many variables, a topic that has been taboo so far. Later in the course, we will develop a theory of continuity, differentiability, and derivative computations for such functions.

4.3. **Dimension, codimension, and intersection theory.** Suppose we are in a vector space of dimension n. We are given subsets M and P which are m-dimensional and p-dimensional respectively, where both m and p are less than or equal to n. What should we *expect* regarding the dimension of the intersection of the two subsets?

Intersection is a top-down approach, so we need to rethink of the subsets in terms of the constraints imposed to get them. The m-dimensional subset M arose because of $n - m$ independent constraints. The number $n - m$, i.e., the difference in dimension of the whole space and the subset, is termed the *codimension* of the subset. Similarly, the subset P is defined using $n - p$ independent constraints, so its codimension is $n - p$.

When we take intersections, then, *generically*, the codimensions add, i.e., the number of constraints needed to define the intersection is, generically, the sum of the number of constraints needed to define each subset. This need not always be true, but it is what we should *expect* in general. This means that the codimension of the intersection $M \cap P$ is $(n - m) + (n - p) = 2n - m - p$ and the dimension is therefore $m + p - n$. If $m + p < n$, then we *should expect, generically*, that the intersection is empty.

Here are some numerical examples:

- In 20-dimensional space, the intersection of an 18-dimensional subset and a 17-dimensional subset is expected to be 15-dimensional: The 18-dimensional subset has codimension 2, the 17-dimensional subset has codimension 3. When we intersect, the codimensions are expected to add, and we get codimension 5, which means dimension 15.
- In 5-dimensional space, the intersection of a 3-dimensional subset and a 2-dimensional subset is expected to be 0-dimensional, and thus is expected to be a point or finite collection of points: The 3-dimensional subspace has codimension 2, and the 2-dimensional subspace has codimension 3. When we intersect, the codimensions are expected to add, so we get $2 + 3 = 5$, so its dimension is 0.
- In 7-dimensional space, the intersection of a 3-dimensional subset and a 2-dimensional subset is expected to be empty, because its expected dimension comes out to be negative: The 3-dimensional subset has codimension 4, and the 2-dimensional subspace has codimension 5, so the codimensions add, and we get $4 + 5 = 9$. But codimension 9 in a 7-dimensional space yields a negative value for the dimension, which is not possible, so the intersection is likely empty.

Note that *in reality*, the intersection may have bigger dimension than expected, which happens if the constraints describing the two subsets are not independent of each other. Also, it is possible that the intersection is a lot smaller, or is even empty. So, while the above is a good thumb rule of what to expect, in practice you actually need to set up equations and solve.

4.4. **Curves in the plane.** A curve in the plane is something 1-dimensional in a 2-dimensional space, so both its dimension and codimension are 1. It could be described in either of these ways:

- The *top-down description* of the curve uses a single equational constraint in $\mathbb{R}^2$. This is a *relational* or *implicit* description, something of the form $F(x, y) = 0$ for some function F of two variables.
- The *bottom-up description* of the curve uses a *parametric* description of the form $x = f(t)$, $y = g(t)$.

We now note that the *intersection* of two curves in the plane is *expected* to be zero-dimensional, which means it is expected to be a finite collection of points. It may well happen that this intersection is zero-dimensional.

How do we *find* the point of intersection? We make various cases:

- If the two curves are given by relational descriptions $F_1(x, y) = 0$ and $F_2(x, y) = 0$, then we try to solve the system of two equations in two variables.
- If one curve is given by the relational description is $F(x, y) = 0$ and the other curve is given by the parametric description is $x = f(t)$, $y = g(t)$, then to solve this, we solve $F(f(t), g(t)) = 0$ as one equation in one variable. For each t-value we get, we compute the corresponding values of x and y.
- If both curves are parametric, say the first curve is given by $x = f_1(t)$, $y = g_1(t)$, and $x = f_2(t)$, $y = g_2(t)$, then to find the intersection, we first make sure that the dummy parameter letters are different. So we rewrite the parameter for the first curve as t_1 and the parameter for the second curve as t_2. We thus have $x = f_1(t_1)$, $y = g_1(t_1)$, and $x = f_2(t_2)$, $y = g_2(t_2)$. To find the intersection, solve the system of two equations: $f_1(t_1) = f_2(t_2)$ and $g_1(t_1) = g_2(t_2)$ in the two variables t_1 and t_2. After finding t_1 and t_2, plug in the function values to find the points x and y.

4.5. **Collision of curves.** The above described the *intersection* of curves. To intersect curves, we are simply interested in where their paths cross. There is a related notion of *collision* of curves with a time parameter, which occurs if two curves intersect with the *same value of time parameter for both.*

In symbols, if $x = f_1(t)$, $y = g_1(t)$ is one parametric curve and $x = f_2(t)$, $y = g_2(t)$ is another parametric curve, then to find whether they collide, we need to solve the system $f_1(t) = f_2(t)$ and $g_1(t) = g_2(t)$. This is a *system of two equations in one variable.*

What does this mean? If the number of equations is more than the number of variables, then we should in general expect *no solutions.* Specifically, in our case, we should not in general expect that the (usually finite) set of solutions to the equation $f_1(t) = f_2(t)$ has a non-empty intersection with the (usually finite) set of solutions to the equation $g_1(t) = g_2(t)$.

However, this *could* happen, even if unexpected.

Collision is much stronger and rarer than intersection, because collision requires the same value of time parameter where the curves intersect.

4.6. **Space curves.** A space curve, or a curve in space, is a 1-dimensional subset in 3-dimensional space. In particular, it has codimension 2. It could be described in two ways:

- A *top-down description*, which involves setting up *two* equations in the three coordinates, i.e., equations of the form $F(x, y, z) = 0$ and $G(x, y, z) = 0$.
- A *bottom-up description*, which involves writing all coordinates as functions of a parameter, i.e., $x = f(t)$, $y = g(t)$, and $z = h(t)$.

What can we say about the *intersections* of space curves? In general, space curves are not expected to intersect, because the codimension of the intersection turns out to be 4, which is bigger than the dimension of the space. However, they just might intersect. Here's how we compute the intersections:

- Suppose the two curves are both given in relational (top-down) form. Say one curve is given by $F_1(x, y, z) = 0$ and $G_1(x, y, z) = 0$ and the other curve is given by $F_2(x, y, z) = 0$ and $G_2(x, y, z) = 0$. To find the intersection, we need to solve the *four* equations $F_1(x, y, z) = 0$, $F_2(x, y, z) = 0$, $G_1(x, y, z) = 0$, $G_2(x, y, z) = 0$ in *three* variables. In general, we do not expect any solutions, because the dimension of the solution space appears to be -1. However, it may happen by sheer chance that there is a solution, i.e., that the curves intersect.
- Suppose one curve is given in relational form and the other curve is given in parametric form. Say, the top-down description for the first curve is $F_1(x, y, z) = 0$ and $G_1(x, y, z) = 0$, and the bottom-up description for the second curve is $x = f_2(t)$, $y = g_2(t)$, and $z = h_2(t)$. To solve these, we plug in the parametric description into the relational description, so we're trying to solve the two equations $F_1(f_2(t), g_2(t), h_2(t)) = 0$ and $G_1(f_2(t), g_2(t), h_2(t)) = 0$. This is s system of *two* equations in one variable, and we *expect* no solutions, because the solution space is expected to have dimension -1. But there may be a solution by chance.

- Suppose both curves are given in parametric form, say the first one is given as $x = f_1(t)$, $y = g_1(t)$, and $z = h_1(t)$. The second curve is given by $x = f_2(t)$, $y = g_2(t)$, and $z = h_2(t)$. In order to find the intersection, we change the dummy parameters to different letters, say t_1 for the first curve and t_2 for the second curve. We now have to solve the three equations $f_1(t_1) = f_2(t_2)$, $g_1(t_1) = g_2(t_2)$, and $h_1(t_1) = h_2(t_2)$ in the two variables.

4.7. **Collision of space curves.** We say that space curves collide if they intersect with the same value of the parameter on both. Note that collision of space curves is even less likely than intersection – the expected dimension of the space of intersection points is -1, the expected dimension of the space of collision points is -2.

Given the parametric description $x = f_1(t)$, $y = g_1(t)$, $z = h_1(t)$ and the parametric description $x = f_2(t)$, $y = g_2(t)$, $z = h_2(t)$. Then, to find the collision points, we need to solve the *three* equations $f_1(t) = f_2(t)$, $g_1(t) = g_2(t)$, $h_1(t) = h_2(t)$ in the *one* variable t.

5. Differentiation of vector-valued functions

5.1. **Definition of derivative.** Given a vector-valued function of the form:

$$f = t \mapsto \langle f_1(t), f_2(t), \dots, f_n(t) \rangle$$

We define:

$$f'(t) = \langle f_1'(t), f_2'(t), \dots, f_n'(t) \rangle$$

In other words, the derivative of a vector-valued function is the vector of the derivatives of each of its scalar functions.

Another equivalent definition is as a *limit of a difference quotient*:

$$f'(t) := \lim_{h \to 0} \frac{f(t+h) - f(t)}{h}$$

Here, h is a *real number*, the subtraction $f(t+h) - f(t)$ is *vector subtraction*, and the division by h is *scalar multiplication* of the scalar $1/h$ by the vector $f(t+h) - f(t)$.

Unpacking this definition gives the same as the definition in terms of coordinates given earlier.

5.2. **Definition of higher derivatives.** Differentiating once was the hard part, now we can just keep going on and on repeatedly. The k^{th} derivative of a vector-valued function is the vector of the k^{th} derivatives of each of its component scalar functions.

In other words, if we have $f(t) = \langle f_1(t), f_2(t), \dots, f_n(t) \rangle$, then:

$$f^{(k)}(t) = \langle f_1^{(k)}(t), f_2^{(k)}(t), \dots, f_n^{(k)}(t) \rangle$$

5.3. **Rule for addition and scalar multiplication.** If f and g are n-dimensional vector-valued functions, then $f + g$ is defined as the n-dimensional vector-valued function that, at each point t, is $f(t) + g(t)$. In turn, to add $f(t) + g(t)$, we add them coordinate-wise. Thus, we're being doubly wise: we're first adding the functions point-wise, and then at each point, we're adding them coordinate-wise.

If both f and g are differentiable, then so is $f + g$, and $(f + g)' = f' + g'$, and if a is a real number, then $(af)' = a(f')$.

This rule extends to higher derivatives as well (secret reason: a composition of linear maps is linear), so we get that if f and g are k times differentiable, then $(f + g)^{(k)} = f^{(k)} + g^{(k)}$ for all natural numbers k, and $(af)^{(k)} = af^{(k)}$ for all real numbers a.

5.4. **Rule for scalar multiplication.** If f is a differentiable scalar-valued function and g is a differentiable vector-valued function, then fg is a differentiable vector-valued function, and:

$$(fg)' = (f')(g) + (f)(g')$$

This is the scalar-vector multiplication version of the product rule. Is it a coincidence that it looks just like the product rule for scalars? No. In fact, there are two far-reaching reasons why any product rule should look like the product rule you are familiar with. We will not cover either right now, but they will both become clear to you later in life.

5.5. **Rule for dot product.** If f and g are both differentiable *as vector-valued functions*, then $f \cdot g$ is differentiable as a scalar-valued function, and we have:

$$(f \cdot g)' = ((f') \cdot g) + (f \cdot (g'))$$

where the addition on the right side is pointwise addition of scalar-valued functions.

5.6. **Rule for cross product.** It turns out that we have a product rule for this, just as we would expect:

$$(f \times g)' = (f' \times g) + (f \times g')$$

However, you need to be careful here, because the cross product is *not commutative.* For the product rule with scalars (and even for the dot product) it is not critical to remember the order in which we write the terms within each product. But with the cross product, it is. *Remember that the order in which the functions being crossed is the same* on the left side and in each of the things being added on the right side.

6. Tangent vectors

6.1. **Tangent vector in the context of a parameterization.** For a curve with parametric description:

$$t \mapsto (f_1(t), f_2(t), \dots, f_n(t))$$

The tangent vector to the curve is the derivative of the corresponding vector-valued function, i.e., the tangent vector at a point t_0 is (as a free vector) $\langle f_1'(t_0), f_2'(t_0), \dots, f_n'(t_0)\rangle$. As a *localized* vector, it is the vector from the point $\langle f_1(t_0), f_2(t_0), \dots, f_n(t_0)\rangle$ to the point $\langle f_1(t_0) + f_1'(t_0), f_2(t_0) + f_2'(t_0), \dots, f_n(t_0) + f_n'(t_0)\rangle$. In other words, it starts at the point where we are taking the tangent vector, and goes the tangent vector.

6.2. **Unit tangent vector to a curve.** It's worth noting that the tangent vector depends on the parameterization, but the *line* along which that vector appears does not. If we choose a different parameterization, then the length of the tangent vector at a point might change. If our parameterization traverses the same curve in reverse, then the direction may become opposite to what it originally was. However, the line of the tangent vector remains the same, i.e., any two tangent vectors for different parameterizations are parallel.

Given this, it makes sense, just given the curve, to talk of the unit tangent vectors (two of them, opposite to each other) and these are independent of the parameterization. Given a direction of traversal (without a concrete parameterization) we can pick the unit tangent vector in the forward direction for that traversal.

7. Integration of vector-valued functions

7.1. **Indefinite integration.** Suppose f is a vector-valued function. We say that F is an *antiderivative* or *indefinite integral* of f if $F' = f$. To find an antiderivative of a vector-valued function, we simply find an antiderivative of each of the coordinate functions.

7.2. **Definite integration.** The *definite integral* of a vector-valued function of t from $t = a$ to $t = b$ is defined coordinate-wise: we perform definite integrations for each of the coordinate function. Specifically if we have:

$$f(t) := \langle f_1(t), f_2(t), \dots, f_n(t)\rangle$$

then the integral is:

$$\int_a^b f(t)\,dt = \langle \int_a^b f_1(t)\,dt, \int_a^b f_2(t)\,dt, \dots, \int_a^b f_n(t)\,dt\rangle$$

This definite integral is thus a n-dimensional vector. Recall that ordinarily, the definite integral of a scalar-valued function between fixed limits is a *number*, i.e., a *scalar*.

7.3. **Fundamental theorem of calculus.** The fundamental theorem of calculus applies for vector integration. This isn't deep – the basic reason here is that it applies in each coordinate.

7.4. **Sums and scalar multiples.** Both indefinite and definite integrals are *linear*, in the sense that they split across sums and scalar multiples can be pulled out of expressions. In particular, we have the following for vector-valued functions f and g, real numbers a and b, and scalars λ:

$$\begin{aligned}\int_a^b ((f(t) + g(t))\,dt &= \int_a^b f(t)\,dt + \int_a^b g(t)\,dt \\ \int_a^b \lambda f(t)\,dt &= \lambda \int_a^b f(t)\,dt\end{aligned}$$

7.5. **Products and chains.** It is possible to write down vector versions of the integration by parts, u-substitution rules, etc. In practice, though, it is much easier, and more general as well as more powerful, to simply do the integrations in each coordinate separately and bring to bear in each coordinate the entire arsenal of techniques for integration in one variable.