

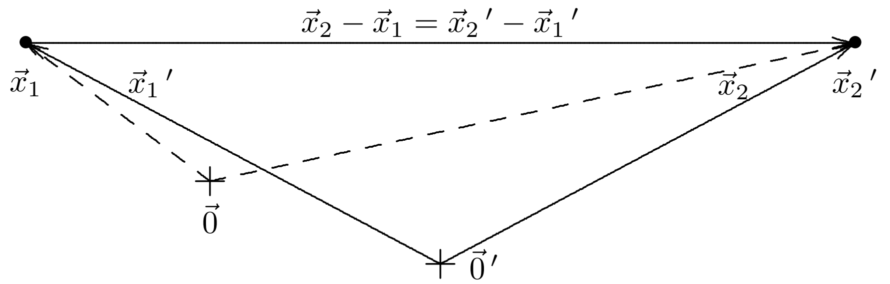
1.3 Points: A Deeper Look

In the first section of this book we listed various types of physical vectors: forces, velocities, and so on. Conspicuously absent from that list of vectorial physical quantities was *position*, the most fundamental of them all. The reason is that there is a special subtlety to the concept of a position vector, which would have made it a misleading example then.

At the beginning of the second section it was observed that once the origin of coordinates is fixed, a point in space (in other words, a possible value for a position variable) can be associated with a vector with tail at the origin. The rest of that section, therefore, was written as if points and vectors are basically the same thing.

Another kind of vector with the same physical units (e.g., meters) as position is *displacement*, the difference between two positions. If two particles are located at positions $\vec{x}_1$ and $\vec{x}_2$ then the second particle is displaced from the first by the vector $\vec{x}_2 - \vec{x}_1$. More generally, the vector difference can be interpreted as the displacement, or relative position, of one geometrical point relative to another point, even when the points are not occupied by physical particles. In the parametric equation of a line, $\vec{x} = t\vec{u} + \vec{x}_0$, $\vec{u}$ and $t\vec{u}$ are displacement vectors, while $\vec{x}_0$ and $\vec{x}$ are position vectors in the absolute sense.

This distinction is not just conceptual; there is a real mathematical difference between the two kinds of vectors. When we use $\mathbf{R}^3$ to model physical space, the location of the origin is chosen arbitrarily. (Of course, in any particular application or calculation, some choices may be more convenient or natural than others.) If the origin is changed, then *the position vectors representing the points $\vec{x}_1$ and $\vec{x}_2$ change.* However, the displacement vector $\vec{x}_2 - \vec{x}_1$ stays the same.



Similarly, vectors of velocity, force, etc. do not depend on the choice of origin. Vectors of each such type belong to their own space, which is not the same as the physical space. The origin of velocity space is the condition of being at rest, which has nothing to do with being located at a particular point. In fact, in pictorial representations one usually thinks of the origins of the spaces of possible velocities, forces, etc. of a particle as being located *at the location of the particle concerned*, not at the origin of coordinates. That is, such vectors should be drawn with their tails at the particle (as indicated in one of the sketches in the next section). This is important for visualizing the effects of such velocities and forces in generating displacements from that point (see the next section).

In some books, position vectors are called *bound vectors*, and vectors that are independent of the coordinate origin are called *free vectors*.

The distinction between two types of vectors has implications for the algebraic operations on vectors. The sum of a position vector and a displacement is a new position vector. The sum of two displacements is another displacement. But to add two position vectors (or to add two points in space to each other) is meaningless.

A very similar situation arises in computer programming in connection with *pointers*, which label locations in the computer's memory.* The sketch shows a model of a computer memory.

1	2	$\cdots$	$2^{32}-1$

"A pointer is a variable that contains the address of another variable." At one level it is just a number. However, the computer language needs to make a rigid distinction between pointers and ordinary integers. A pointer is like a position vector; the difference between two pointers is an integer (which can be used, for instance, as a subscript of an array) and is like a displacement vector. It is therefore legal to subtract two pointers (getting an integer) or to add or subtract an integer from a pointer (getting a new pointer), but attempting to add two pointers should produce an error message. Pointers, and only they, depend on "origin": The pointers locating an array (subscripted variable) in memory may change as the program is revised, or executed under different circumstances, but the integer subscript that indexes a particular element of the array will stay the same.

Returning to physical vectors, observe that the change of origin is unlike other changes of coordinates, such as a rotation of axes around the origin, or a change in the units in which the coordinates are measured from meters to feet. Coordinate (or basis) changes of the latter type (which will be studied extensively in Chapter 4) change the *numerical representation* of a vector as a string of numbers, but leave the vector itself, as an abstract object, unchanged (and there is no difference between free and bound vectors in these respects). The discussion above shows that for *points* there is another necessary level of abstraction: The representation of points by vectors is somewhat arbitrary (depending on the choice of origin), just as the representation of physical vectors by n-tuples of numbers is arbitrary (depending on the direction and scale of the axes).

There are physical situations where the space of possible positions is not "flat" (such as the space-time of general relativity, or the space of possible orientations of a rigid body). In such cases the points cannot be regarded as vectors at all. Nevertheless, velocities (more generally, tangents to curves — see the next section) are still correctly modeled by vectors.

* B. W. Kernighan and D. M. Ritchie, *The C Programming Language*, Prentice–Hall, Englewood Cliffs, 1978, Chapter 5.

This is part of the subject called *differential geometry*, for which the material in this book is an important prerequisite.

Length and distance

This is a good opportunity to review some elementary terminology and notation that didn't make their way into the previous sections.

Numbers (real or complex) are called *scalars* when it is desired to distinguish them from vectors. (This terminology carries over to *functions* that take scalar or vector values, respectively.) The operation of multiplying a vector by a number is called *scalar multiplication* to distinguish it from other kinds of multiplication involving vectors (such as the dot and cross products).

In the spaces $\mathbf{R}^n$ and the spaces of physical quantities modeled by them, each vector has a *length* defined by

$$\|\vec{v}\| = \sqrt{\vec{v}\cdot\vec{v}} = \sqrt{\sum_{j=1}^{n} |v_j|^2}\,.$$

(The notation $|\vec{v}|$ is also used.) This scalar tells the "size" of the vector, discarding the directional information.

The *distance* between two vectors, $\vec{u}$ and $\vec{v}$, is the length of their difference vector, $\|\vec{u}-\vec{v}\|$. When the two vectors represent points, their difference is a displacement vector and hence the distance is independent of the choice of origin.

Exercises

1.3.1 Which of the following operations make sense, and which kind of vector is the result in each case?

(a) Subtraction of one position vector from another.

(b) Subtraction of a displacement vector from a position vector.

(c) Subtraction of one displacement vector from another.

(d) Subtraction of a position vector from a displacement vector.

1.3.2 Let $\vec{x}_1 = 2\hat{\imath}+\hat{\jmath}-3\hat{k}$, $\vec{x}_2 = \hat{\imath}+\hat{\jmath}+\hat{k}$. Calculate $\|\vec{x}_1\|$, $\|\vec{x}_2\|$, and $\|\vec{x}_2-\vec{x}_1\|$. (Remember that $2\hat{\imath}+\hat{\jmath}-3\hat{k}$ means the same thing as $(2,1,-3)$.)

1.3.3 Redefine coordinates in $\mathbf{R}^3$ by

$$x' = x-1, \quad y' = y+2, \quad z' = z.$$

Calculate the primed coordinates of the vectors $\vec{x}_1$ and $\vec{x}_2$ of the previous exercise, and verify that $\vec{x}_2-\vec{x}_1$ and $\|\vec{x}_2-\vec{x}_1\|$ are the same when calculated in the primed coordinates as in the original coordinates, although the numerical values of $\|\vec{x}_1\|$ and $\|\vec{x}_2\|$ change.