Kinematics-Chapter-3.nb

Written in *Mathematica* 12.2.0.0 Vince Cronin
Begun February 28, 2021, with some material from earlier documents; revised March 3.

3.1 Specifying a vector in different 2D coordinate systems

3.1.1 Introduction to the problem

Asserting that something is moving leads immediately to the question “moving relative to what?” The quantitative description of motion requires us to understand coordinate systems and how coordinate systems relate to one another. The process of gaining that understanding typically begins by working through a simple problem, in which a given vector is defined in an initial coordinate system and then the same vector is re-defined in a second coordinate system identical in all respects to the first except that the second is rotated around their common origin. This type of problem involves coordinate transformation. I have adapted material from a vector analysis textbook by Harry Davis and Arthur Snider (1987) for this chapter.

3.1.2 How do we determine the coordinates of a vector in a 2D coordinate system that is rotated from the vector’s original 2D coordinate system?

Imagine a point Q whose location is specified by vector q in a right-orthogonal Cartesian X-Y coordinate system (Figure 3.1). The coordinates of q in the X-Y coordinate system are $\{1, 2\}$. The length of vector q (symbolized by $[\![q]\!]$) is

$$[\![q]\!] = \sqrt{1^2 + 2^2} \simeq 2.24$$

The length of vector q does not vary in other right-orthogonal Cartesian coordinate systems that share the same scale as the X-Y coordinate system in which q was defined.

Now, we will specify a second coordinate system, called the X’-Y’ system, with the same origin as the X-Y coordinate system. The X’ axis is oriented θ=30° in a positive or anti-clockwise direction from the X axis, and the Y’ axis is θ=30° from the Y axis. What are the coordinates of point Q in the X’-Y’ coordinate system?

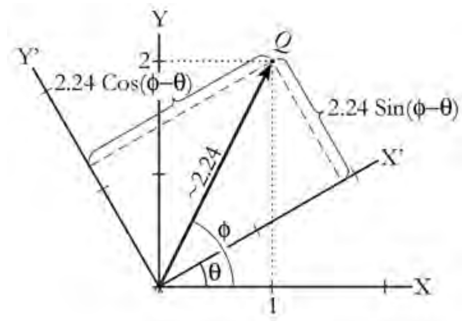

Figure 3.1. Vector to point Q is q = {1, 2} in the 2-dimensional X-Y coordinate system. The X'-Y' coordinate system has the same origin as the X-Y coordinate system but the X' axis is oriented θ ° anticlock-wise (that is, +30°) from the X axis. The length or magnitude of q is ~2.24.

One way to solve this problem is to carefully draft a scaled picture of the vector q and the two coordinate systems (Fig. 3.1), and use this to compute or measure the coordinates of Q in the X'-Y' coordinate system. By solving a right-triangle problem, we can see that the coordinate of Q in the X' direction is ($[\![q]\!]$ cos(ϕ–θ)) and the Y' coordinate is ($[\![q]\!]$ sin(ϕ–θ)). We will define vector q' as the vector to point Q in the X'-Y' coordinate system, where

q' = {($[\![q]\!]$ cos(ϕ–θ)), ($[\![q]\!]$ sin(ϕ–θ))}, where

$\phi = \tan^{-1}\left(\frac{2}{1}\right) \simeq 63.4°$

q' = {1.86603, 1.23205}.

That was a lot of work. While we discussed this problem as a 2-dimensional problem involving the X-Y coordinate plane, we can add the third dimension by noting that the Z coordinate axis extends up from the origin of the coordinate system and perpendicular to both the X and Y coordinate axes. The Z' coordinate axis coincides with the Z coordinate axis, so all of the differences between these two coordinate systems can be seen in the X-Y plane — the same as the X'-Y' plane. In three dimensions,

q = {1, 2, 0}, and

q' = {($[\![q]\!]$ cos(ϕ–θ)), ($[\![q]\!]$ sin(ϕ–θ)), 0}.

Thankfully there is another and (ultimately) simpler way of doing this kind of problem. But before we get to the simpler way, we need to learn a little bit about matrices.

3.2 A taste of matrix mathematics

3.2.1 What is a matrix?

A matrix is a rectangular array of numbers contained within brackets. A vector can be represented as a matrix, and is usually shown as a *column matrix* because it has a vertical array of numbers. For example, we can represent vector a in matrix form as

$$a = \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix}$$

The subscript numbers indicate what row each element occupies. We can define a matrix b that has 3 rows and 3 columns — called a *square matrix* because it has the same number of rows and columns.

$$b = \begin{pmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{pmatrix}$$

The pair of subscript numbers indicate what row and column each element occupies. For example, the element on the third row and second column is designated b_{32}.

In *Mathematica*, input lines involving matrices can be formatted in two ways: in *list form* (that is, as lists or nested lists) or in *matrix form*.
In list form, the matrix associated with vector a can be input as
a={a_1,a_2,a_3} with curly brackets and no intervening spaces, and the matrix b can be input as a set of nested lists with the structure b={{b_{11},b_{12},b_{13}},{b_{21},b_{22},b_{23}},{b_{31},b_{32},b_{33}}} or {{FirstRow},{S-econdRow},{ThirdRow}}.

Example. Here are examples of input lines defining matrices a and b in list form, with numbers filling each of the matrix positions.

```
a={11,15,18};
b={{0.2,0.6,0.5},{0.8,0.9,0.7},{0.4,0.1,0.3}};
```

We can use the Basic Math Assistant palette (which is accessible from the Palettes drop-down menu along the top of your screen while the *Mathematica* application is active) to define matrices for input lines. Insert the cursor in the appropriate spot in your Mathematica notebook. Then open the Typesetting part of the Basic Math Assistant palette, select the "typesetting forms" (the button on the left end of the header bar in the Typesetting section), and click on the symbol that looks like this:

$$\begin{pmatrix} \square & \square \\ \square & \square \end{pmatrix}$$

Extra columns are created by inserting the cursor inside the parentheses in the default 4x4 matrix, then pressing [CTRL ,] (that is, press the comma key with the control key depressed), and extra rows are created by pressing [CTRL RET]. When you get a matrix template that looks like this,

$$\begin{pmatrix} \square & \square & \square \\ \square & \square & \square \\ \square & \square & \square \end{pmatrix},$$

you can click on each individual square to insert a value or variable name in that space. Follow the same process to make a column matrix, except begin the process by selecting the symbol that looks like this:

$$\begin{pmatrix} \square \\ \square \end{pmatrix}.$$

Example. Here are examples of input lines defining matrices a and b in matrix form, with numbers filling each of the matrix positions.

```
a= (11
    15
    18);
b= (0.2 0.6 0.5
    0.8 0.9 0.7
    0.4 0.1 0.3);
```

3.2.2 How do we multiply a column matrix by a square matrix?

We can multiply column matrix a and square matrix b together to form a column matrix c (that is, $c = b \cdot a$), or

$$\begin{pmatrix} c_1 \\ c_2 \\ c_3 \end{pmatrix} = \begin{pmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix}$$

We can think of each of the three rows of square matrix b as *row vectors* (that is, a row of a matrix used as a vector).

TopRowVector $= \{b_{11}, b_{12}, b_{13}\}$
MiddleRowVector $= \{b_{21}, b_{22}, b_{23}\}$
BottomRowVector $= \{b_{31}, b_{32}, b_{33}\}$

$$b = \begin{pmatrix} \text{TopRowVector} \\ \text{MiddleRowVector} \\ \text{BottomRowVector} \end{pmatrix}$$

We can think of the matrix multiplication $c = b \cdot a$ as

$$\begin{pmatrix} c_1 \\ c_2 \\ c_3 \end{pmatrix} = \begin{pmatrix} (\text{TopRowVector} \cdot a) \\ (\text{MiddleRowVector} \cdot a) \\ (\text{BottomRowVector} \cdot a) \end{pmatrix} = \begin{pmatrix} (b_{11}\, a_1) + (b_{12}\, a_2) + (b_{13}\, a_3) \\ (b_{21}\, a_1) + (b_{22}\, a_2) + (b_{23}\, a_3) \\ (b_{31}\, a_1) + (b_{32}\, a_2) + (b_{33}\, a_3) \end{pmatrix}$$

Each element of c is the result of the dot product of one of the matrix b row vectors and vector a. The product of multiplying a column matrix with three rows (vector a) by a 3-by-3 square matrix (matrix b) is a column matrix with three rows (vector c).

Example. Find the result of the following matrix multiplication:

$$\begin{pmatrix} 0.2 & 0.6 & 0.5 \\ 0.8 & 0.9 & 0.7 \\ 0.4 & 0.1 & 0.3 \end{pmatrix} \begin{pmatrix} 11 \\ 15 \\ 18 \end{pmatrix}$$

Solution

$$\begin{pmatrix} 0.2 & 0.6 & 0.5 \\ 0.8 & 0.9 & 0.7 \\ 0.4 & 0.1 & 0.3 \end{pmatrix} \begin{pmatrix} 11 \\ 15 \\ 18 \end{pmatrix} = \begin{pmatrix} (0.2 * 11) + (0.6 * 15) + (0.5 * 18) \\ (0.8 * 11) + (0.9 * 15) + (0.7 * 18) \\ (0.4 * 11) + (0.1 * 15) + (0.3 * 18) \end{pmatrix} = \begin{pmatrix} 20.2 \\ 34.9 \\ 11.3 \end{pmatrix}$$

The product of the two matrices is the following 3-component vector: {20.2, 34.9, 11.3}.

In *Mathematica*, the corresponding code looks like the following, in which we define **vectorA** and **matrixB**

```
vectorA = (11
           15
           18);
matrixB = (0.2 0.6 0.5
           0.8 0.9 0.7
           0.4 0.1 0.3);
```

and then define vectorC as the dot product of these two matrices in the correct order with **matrixB** before **vectorA**.

```
vectorC = matrixB.vectorA;
```

The period between **matrixB** and **vectorA** indicates a dot product. The numerical result, displayed in matrix form using the built-in *Mathematica* functions **N** and **MatrixForm[]**, is

```
MatrixForm[N[vectorC]]
```

$$\begin{pmatrix} 20.2 \\ 34.9 \\ 11.3 \end{pmatrix}$$

Alternatively, we could have found **vectorC** by using the built-in *Mathematica* function **Dot[]** as follows.

```
vectorC = Dot[matrixB,vectorA]
```

and the numerical result in matrix form is

```
MatrixForm[N[vectorC]]
```

$$\begin{pmatrix} 20.2 \\ 34.9 \\ 11.3 \end{pmatrix}$$

3.3 The transformation matrix and its inverse

3.3.1 A bit of background

First, let's find the unit vector along the X axis and call it $\hat{i}$, which is often called "i-hat" since the diacritic circumflex symbol (^) above the *i* looks vaguely like a hat. We will use the ^ over a vector symbol to indicate that it is a unit vector. Next, we'll find the unit vector along the Y axis (yielding unit vector $\hat{j}$), Z axis ($\hat{k}$), X' axis ($\hat{i}'$), Y' axis ($\hat{j}'$) and Z' axis ($\hat{k}'$). We can define a coordinate transformation matrix *J* given by

$$J = \begin{pmatrix} \hat{i}' \cdot \hat{i} & \hat{i}' \cdot \hat{j} & \hat{i}' \cdot \hat{k} \\ \hat{j}' \cdot \hat{i} & \hat{j}' \cdot \hat{j} & \hat{j}' \cdot \hat{k} \\ \hat{k}' \cdot \hat{i} & \hat{k}' \cdot \hat{j} & \hat{k}' \cdot \hat{k} \end{pmatrix}$$

where the 9 elements of the matrix are each dot products of unit vectors along coordinate axes. Geometrically, $\hat{i}' \cdot \hat{i}$ is the cosine of the angle between the X and X' axes (that is, between $\hat{i}'$ and $\hat{i}$). This is also called a *direction cosine* in this application.

The inverse of transformation matrix J is

$$J^{-1} = \begin{pmatrix} \hat{i}' \cdot \hat{i} & \hat{j}' \cdot \hat{i} & \hat{k}' \cdot \hat{i} \\ \hat{i}' \cdot \hat{j} & \hat{j}' \cdot \hat{j} & \hat{k}' \cdot \hat{j} \\ \hat{i}' \cdot \hat{k} & \hat{j}' \cdot \hat{k} & \hat{k}' \cdot \hat{k} \end{pmatrix}$$

Multiplication of a square matrix like J by its inverse J^{-1} yields the identity matrix, which is composed of 1s along the diagonal and 0s in the off-axis positions. If a matrix is multiplied by its inverse, the result is the identity matrix whose diagonal terms are all 1s and whose off-axis terms are all 0s.

$$J \cdot J^{-1} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Transformation matrix J is an orthogonal matrix, so its inverse is the same as its transpose. What is the transpose of a matrix? If we define a matrix M as follows,

$$M = \begin{pmatrix} A & d & e \\ g & B & h \\ n & -f & C \end{pmatrix}$$

the *transpose* of matrix M is represented by M^T and is

$$M^T = \begin{pmatrix} A & g & n \\ d & B & -f \\ e & h & C \end{pmatrix}.$$

The values along the diagonal of the transposed matrix M^T are unchanged from the original matrix M, but the values across the diagonal from each other are swapped. The *inverse* of a matrix M is indicated by M^{-1}. If a matrix is multiplied by its inverse, the result is the identity matrix whose diagonal terms are all 1s and whose off-axis terms are all 0s.

$$M \cdot M^{-1} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Finally, if the transpose of a matrix is the same as the inverse of the matrix (that is, if $M^T = M^{-1}$), that matrix (M) is called an *orthogonal* matrix.

3.4 Transforming a vector from one coordinate system to another

3.4.1 The good news

If you know the coordinates of a vector in one coordinate system (X-Y), you can multiply that vector by the appropriate transformation matrix to obtain the coordinates of that same vector in a different coordinate system (X’-Y’) where both coordinate systems have the same origin. For the 2-D problem we began with, involving vector q, the matrix J transforms the coordinates from the original coordinate system to a new coordinate system that is rotated θ degrees from the original in the same plane (Figure 3-2).

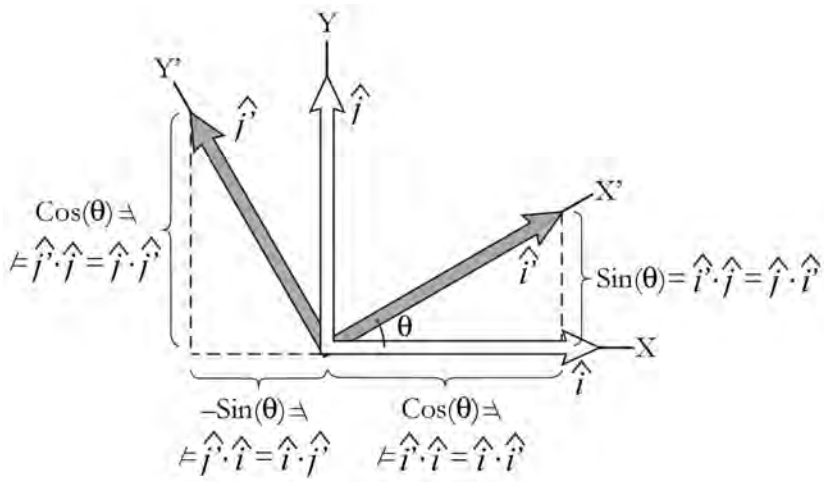

Figure 3.2. Projections of rotated unit vectors onto unit vectors along the axes of the original X - Y coordinate system.

3.4.2 How do we use matrices to transform between coordinate systems in 2 dimensions?

Adapting the 2-dimensional solution from the 3-D solution shown above, we have

$$J_{XY} = \begin{pmatrix} \hat{i}' \cdot \hat{i} & \hat{i}' \cdot \hat{j} \\ \hat{j}' \cdot \hat{i} & \hat{j}' \cdot \hat{j} \end{pmatrix}$$

Another way of visualizing the components of J in 2-D is

$$J_{XY} = \begin{pmatrix} X \text{ coordinate of unit vector } \hat{i}' & Y \text{ coordinate of unit vector } \hat{i}' \\ X \text{ coordinate of unit vector } \hat{j}' & Y \text{ coordinate of unit vector } \hat{j}' \end{pmatrix}$$

If, in our 2-D example, the X’ axis is a positive (counter-clockwise) rotation from the X axis as in Fig. 3-2, the matrix J_{XY} is defined for our purposes as follows:

$$J_{XY} = \begin{pmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{pmatrix}$$

Let’s see if we get the same results as we derived by hand. Vector q in the X-Y coordinate system is defined as

$$q = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$$

and the same vector in the X'-Y' coordinate system is q', where

$q' = J_{XY} \cdot q$, or

$$q' = \begin{pmatrix} \cos(30°) & \sin(30°) \\ -\sin(30°) & \cos(30°) \end{pmatrix} \begin{pmatrix} 1 \\ 2 \end{pmatrix}$$

$$q' = \begin{pmatrix} 1.86603 \\ 1.23205 \end{pmatrix}.$$

That is the same answer we computed longhand in section 3.1.2.

In *Mathematica*, the corresponding code looks like the following, in which we define **vectorQ** and **matrixJ** in the 2-dimensional solution as

```
vectorQ = (1
           2);
theta = 30;
matrixJ = ( Cos[theta Degree]   Sin[theta Degree]
           -Sin[theta Degree]  Cos[theta Degree]);
```

and then define **vectorQPrime** as the dot product of these two matrices in the correct order with **matrixJ** before **vectorQ**.

```
vectorQPrime = matrixJ.vectorQ;
```

The period between **matrixB** and **vectorA** indicates a dot product. The numerical result, displayed in matrix form using the built-in *Mathematica* functions **N** and **MatrixForm[]**, is

```
MatrixForm[N[vectorQPrime]]
```

$$\begin{pmatrix} 1.86603 \\ 1.23205 \end{pmatrix}$$

Alternatively, we could have found **vectorQPrime** by using the built-in *Mathematica* function **Dot[]** as follows.

```
vectorQPrime = Dot[matrixJ,vectorQ]
```

and the numerical result in matrix form is

```
MatrixForm[N[vectorQPrime]]
```

$$\begin{pmatrix} 1.86603 \\ 1.23205 \end{pmatrix}$$

3.4.2 How do we use matrices to transform between coordinate systems in 3 dimensions?

In section 3.1.2, we noted that we could add the third dimension to the original 2-D problem by noting that the Z coordinate axis extends up from the origin of the coordinate system and perpendicular to both the X and Y coordinate axes. The Z' coordinate axis coincides with the Z coordinate axis, so all of the differences between these two coordinate systems can be seen in the X-Y plane — the same as the X'-Y' plane. In three dimensions,

$$q = \begin{pmatrix} 1 \\ 2 \\ 0 \end{pmatrix}$$

The transformation matrix J in 3 dimensions is

$$J = \begin{pmatrix} \hat{i}' \cdot \hat{\imath} & \hat{i}' \cdot \hat{\jmath} & \hat{i}' \cdot \hat{k} \\ \hat{j}' \cdot \hat{\imath} & \hat{j}' \cdot \hat{\jmath} & \hat{j}' \cdot \hat{k} \\ \hat{k}' \cdot \hat{\imath} & \hat{k}' \cdot \hat{\jmath} & \hat{k}' \cdot \hat{k} \end{pmatrix}$$

Another way of visualizing the components of J in 3-D is

$$J = \begin{pmatrix} X \text{ coordinate of } \hat{i}' & Y \text{ coordinate of } \hat{i}' & Z \text{ coordinate of } \hat{i}' \\ X \text{ coordinate of } \hat{j}' & Y \text{ coordinate of } \hat{j}' & Z \text{ coordinate of } \hat{j}' \\ X \text{ coordinate of } \hat{k}' & Y \text{ coordinate of } \hat{k}' & Z \text{ coordinate of } \hat{k}' \end{pmatrix}$$

In this problem, the Z and Z' axes are coincident, so the J matrix is

$$J = \begin{pmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{pmatrix}, \text{ so}$$

$$q' = J \cdot q$$

$$q' = \begin{pmatrix} \cos(30°) & \sin(30°) & 0 \\ -\sin(30°) & \cos(30°) & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 2 \\ 0 \end{pmatrix}$$

$$q' = \begin{pmatrix} 1.86603 \\ 1.23205 \\ 0 \end{pmatrix}$$

In *Mathematica*, the corresponding code looks like the following, in which we re-define **vectorQ** and **matrixJ** in the 3-dimensional solution as

```
vectorQ = (1
           2
           0);
theta=30;
matrixJ = ( Cos[theta Degree]   Sin[theta Degree]  0
           -Sin[theta Degree]   Cos[theta Degree]  0
                0                     0            1);
```

and then re-define **vectorQPrime** as the dot product of these two matrices in the correct order with **matrixJ** before **vectorQ**.

```
vectorQPrime = matrixJ.vectorQ;
```

The period between **matrixJ** and **vectorQ** indicates a dot product. The numerical result, displayed in matrix form using the built-in *Mathematica* functions `N` and `MatrixForm[]`, is

```
MatrixForm[N[vectorQPrime]]
```

$$\begin{pmatrix} 1.86603 \\ 1.23205 \\ 0 \end{pmatrix}$$

Alternatively, we could have found `vectorQPrime` by using the built-in *Mathematica* function `Dot[]` as follows.

```
vectorQPrime = Dot[matrixJ,vectorQ]
```

and the numerical result in matrix form is

```
MatrixForm[N[vectorQPrime]]
```

$$\begin{pmatrix} 1.86603 \\ 1.23205 \\ 0 \end{pmatrix}$$

Exercise 3.1 (HW-05) A vector a is defined as $a = \{1,2,0\}$ and extends to a point A in the XYZ coordinate system.
(a) What are the coordinates of the vector to point A (call it vector a') in a X'Y'Z' coordinate system in which the Z and Z' axes coincide with one another and the X' axis is rotated 47° anti-clockwise from X (and so Y' is 47° from Y). An anti-clockwise rotation is considered a positive rotation.
(b) What are the coordinates of that vector to point A (call it vector a'') in a X"Y"Z" coordinate system in which the Z and Z" axes coincide with one another and the X" axis is rotated 63° clockwise from X (and so Y" is 63° from Y). A clockwise rotation is considered a negative rotation.
Write a *Mathematica* notebook that analyzes the input data to answer questions (a) and (b).

References

Boyd, S., and Vandenberghe, L., 2018, Introduction to Applied Linear Algebra — Vectors, Matrices, and Least Squares: Cambridge, UK, Cambridge University Press, 463 p.

Davis, H.F., and Snider, A.D., 1987, Introduction to Vector Analysis [5th edition]: Boston, Massachusetts, Allyn and Bacon, 365 p.

Weisstein, E.W., Coordinate System: available online via *MathWorld* -- a Wolfram web resource, and accessed 18 January 2012. http://mathworld.wolfram.com/CoordinateSystem.html

Weisstein, E.W., Matrix: available online via *MathWorld* -- a Wolfram web resource, and accessed 18 January 2012. http://mathworld.wolfram.com/Matrix.html

Weisstein, E.W., Matrix Inversion: available online via *MathWorld* -- a Wolfram web resource, and accessed 18 January 2012. http://mathworld.wolfram.com/MatrixInversion.html

Weisstein, E.W., Matrix Multiplication: available online via *MathWorld* -- a Wolfram web resource, and accessed 18 January 2012. http://mathworld.wolfram.com/MatrixMultiplication.html