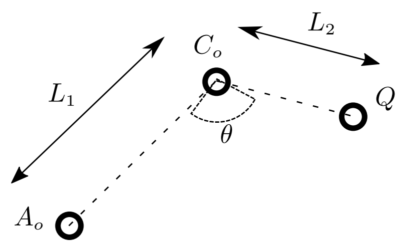
1 Vector Operations With No "Coordinate System"

Euclidean vectors have two properties: **magnitude** and **direction**. We will repeat this over and over in this section!

Consider the graphic below left with points A_o, C_o, and Q. The distance between A_o and C_o is $L_1 = 3$ meters, and the distance between C_o and Q is $L_2 = 2$ meters. The angle θ labeled in the diagram is 120°.

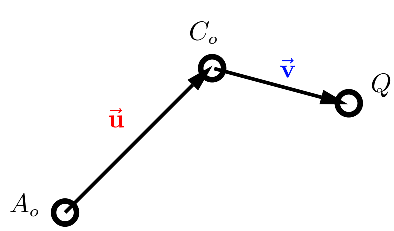
Calculate the distance between A_o and C_o.

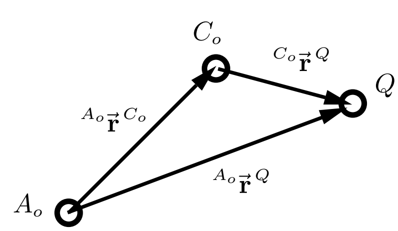


You may be tempted to start doing trigonometry on this picture. Don't do it! The purpose of the development on these pages is to show you how to think in terms of vectors, not trigonometry. It may feel heavy-handed for this simple problem, but thinking in terms of vectors allow a level of abstraction that can make more complex problems tractable and easier to transfer to computer code. This is key in many domains such as graphics, aerospace, and robotics.

First we'll introduce two vectors $\vec{\mathbf{u}}$ and $\vec{\mathbf{v}}$ to "work with" in this problem, drawn on the figure below left. These vectors form a "basis" for the 2D plane of the page, meaning any vector in the page can be expressed as a linear combination of $\vec{\mathbf{u}}$ and $\vec{\mathbf{v}}$. The choice of basis vectors is arbitrary and non-unique, but we've picked these vectors because they are the easiest quantities to work with given the information we have.

1. $\vec{\mathbf{u}}$ has length 3 (magnitude) and points from A_o to C_o (direction).
2. $\vec{\mathbf{v}}$ has length 2 (magnitude) and points from C_o to Q (direction).





Our notation for general position vectors indicates where the "tail" is and where the head (arrow) is. For example, the vector pointing from A_o to Q is ${}^{A_o}\vec{\mathbf{r}}^{Q}$, as shown above right. This notation is "basis independent", meaning it is abstract and uncommitted to how it is expressed. It is, however, very specific about what it **means** geometrically. Later we will see how a basis-independent vector can be expressed using different combinations of other vectors.

Then we can express the basis-independent position vectors between the points above as follows:

$$
\begin{aligned}
{}^{A_o}\vec{\mathbf{r}}^{C_o} &= 1\,\vec{\mathbf{u}} \\
{}^{C_o}\vec{\mathbf{r}}^{Q} &= 1\,\vec{\mathbf{v}} \\
{}^{A_o}\vec{\mathbf{r}}^{Q} &= {}^{A_o}\vec{\mathbf{r}}^{C_o} + {}^{C_o}\vec{\mathbf{r}}^{Q} = 1\vec{\mathbf{u}} + 1\vec{\mathbf{v}}
\end{aligned}
$$

Now to answer the actual question: what is the distance from A_o to Q? The distance is the magnitude of ${}^{A_o}\vec{\mathbf{r}}^{Q}$, which we can find by applying the **defintion** of vector dot product: $\vec{\mathbf{a}} \cdot \vec{\mathbf{b}} \triangleq |\vec{\mathbf{a}}||\vec{\mathbf{b}}| \cos(\angle(\vec{\mathbf{a}}, \vec{\mathbf{b}}))$

$$\left|{}^{A_o}\vec{\mathbf{r}}^{\,Q}\right| \left|{}^{A_o}\vec{\mathbf{r}}^{\,Q}\right| \cancelto{1}{\cos(\angle({}^{A_o}\vec{\mathbf{r}}^{\,Q},\ {}^{A_o}\vec{\mathbf{r}}^{\,Q}))} \quad = \quad {}^{A_o}\vec{\mathbf{r}}^{\,Q} \cdot {}^{A_o}\vec{\mathbf{r}}^{\,Q} \qquad \text{Definition}$$

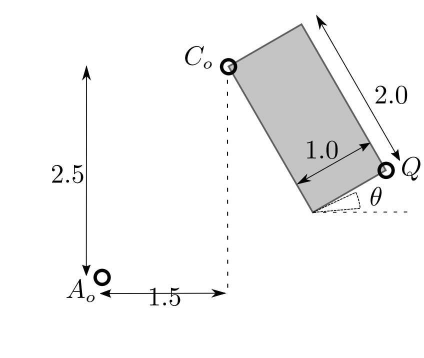
$$\begin{aligned}
\left|{}^{A_o}\vec{\mathbf{r}}^{\,Q}\right|^2 &= (1\vec{\mathbf{u}}\ +\ 1\vec{\mathbf{v}})\cdot(1\vec{\mathbf{u}}\ +\ 1\vec{\mathbf{v}}) && \text{Simplify and Substitute} \\
&= \quad 1 * 1\ (\vec{\mathbf{u}}\cdot\vec{\mathbf{u}}) \quad + 1 * 1\ (\vec{\mathbf{u}}\cdot\vec{\mathbf{v}}) && \text{Distribute and Collect Scalars} \\
&\quad + 1 * 1\ (\vec{\mathbf{v}}\cdot\vec{\mathbf{u}}) \quad + 1 * 1\ (\vec{\mathbf{v}}\cdot\vec{\mathbf{v}}) \\
&= \quad (3 * 3\ \cos\angle(\vec{\mathbf{u}},\vec{\mathbf{u}})) + (3 * 2\ \cos\angle(\vec{\mathbf{u}},\vec{\mathbf{v}})) && \text{Re-apply Dot Product Definition} \\
&\quad + (2 * 3\ \cos\angle(\vec{\mathbf{v}},\vec{\mathbf{u}})) + (2 * 2\ \cos\angle(\vec{\mathbf{v}},\vec{\mathbf{v}})) \\
&= \quad 9\ \cos(0^\circ) \quad + 6\ \cos(60^\circ) && \text{Re-apply Dot Product Definition} \\
&\quad + 6\ \cos(60^\circ) \quad + 4\ \cos(0^\circ) \\
&= \quad 19
\end{aligned}$$

So the distance between A_o and Q is $\sqrt{(19)} \approx 4.36$ meters. If you've studied vector geometry using traditional methods the process above may be somewhat surprising or unsettling.

1. We did not introduce a coordinate system with orthogonal directions, nor did we "resolve components".
2. Our chosen basis vectors were not orthogonal and not unit length.
3. We simply used the magnitude and direction of the vectors we had and applied definitions until we were done.

2 Vectors with Multiple Bases

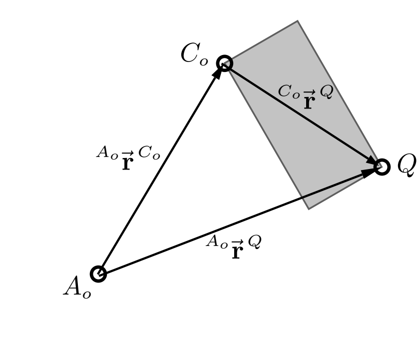
Now consider the picture below of a book hovering above the ground. Which is designed to look similar to the one in the previous section although some dimensions have changed. Relevant measured lengths and angles are shown in the figure.

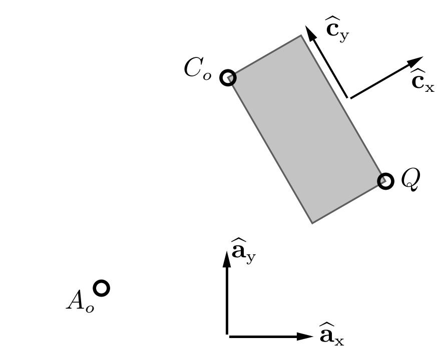


Let's find:

1. the distance between A_o and C_o.
2. the distance between A_o and Q.

The abstract concept is the same as before. We will write basis-independent position vectors (shown below left), choose basis vectors to work with, and proceed with the math.





In the figure above right, we create a rigid vector basis A consisting of mutually orthogonal, unit-length vectors $\widehat{\mathbf{a}}_x$, $\widehat{\mathbf{a}}_y$, $\widehat{\mathbf{a}}_z$. We create a similar basis C of vectors $\widehat{\mathbf{c}}_x$, $\widehat{\mathbf{c}}_y$, $\widehat{\mathbf{c}}_z$ rigidly attached to the book. Basis C is initially aligned with A and then undergoes a right-handed rotation of angle θ about $\widehat{\mathbf{a}}_z = \widehat{\mathbf{c}}_z$ (counter-clockwise in this picture).

We can now write the position vectors as:

$$\begin{aligned}
{}^{A_o}\vec{\mathbf{r}}^{C_o} &= 1.5\,\widehat{\mathbf{a}}_x + 2.5\,\widehat{\mathbf{a}}_y \\
{}^{C_o}\vec{\mathbf{r}}^{Q} &= 1\,\widehat{\mathbf{c}}_x + {}^{-}2\,\widehat{\mathbf{c}}_y \\
{}^{A_o}\vec{\mathbf{r}}^{Q} &= {}^{A_o}\vec{\mathbf{r}}^{C_o} + {}^{C_o}\vec{\mathbf{r}}^{Q} = 1.5\,\widehat{\mathbf{a}}_x + 2.5\,\widehat{\mathbf{a}}_y + 1\,\widehat{\mathbf{c}}_x + {}^{-}2\,\widehat{\mathbf{c}}_y
\end{aligned}$$

(1) Now to find the distance between A_o and C_o:

$$\begin{aligned}
\left|{}^{A_o}\vec{\mathbf{r}}^{C_o}\right|^2 &= {}^{A_o}\vec{\mathbf{r}}^{C_o} \cdot {}^{A_o}\vec{\mathbf{r}}^{C_o} \\
&= (1.5\,\widehat{\mathbf{a}}_x + 2.5\,\widehat{\mathbf{a}}_y)\cdot(1.5\,\widehat{\mathbf{a}}_x + 2.5\,\widehat{\mathbf{a}}_y) && \text{Setup} \\
&= 1.5{*}1.5\,\cancelto{1}{(\widehat{\mathbf{a}}_x\cdot\widehat{\mathbf{a}}_x)} + 1.5{*}2.5\,\cancelto{0}{(\widehat{\mathbf{a}}_x\cdot\widehat{\mathbf{a}}_y)} && \text{Distribute and Collect Scalars} \\
&\quad + 2.5{*}1.5\,\cancelto{0}{(\widehat{\mathbf{a}}_y\cdot\widehat{\mathbf{a}}_x)} + 2.5{*}2.5\,\cancelto{1}{(\widehat{\mathbf{a}}_y\cdot\widehat{\mathbf{a}}_y)} \\
&= 1.5^2 + 2.5^2 = 8.5 && \text{Re-apply Dot Product Definitions and Simplify} \\
\left|{}^{A_o}\vec{\mathbf{r}}^{C_o}\right| &= \sqrt{8.5} \approx 2.92 \text{ meters}
\end{aligned}$$

In this case the magnitude is the square root of the sum of the squares of the coefficients, which should be familiar from calculus or physics. Note **why** it works: it's only because ${}^{A_o}\vec{\mathbf{r}}^{C_o}$ is expressed in terms of vectors that are mutually orthogonal and unit length.

(2) Now to find the distance between A_o and Q:

$$\begin{aligned}
\left|{}^{A_o}\vec{\mathbf{r}}^{Q}\right|^2 &= {}^{A_o}\vec{\mathbf{r}}^{Q} \cdot {}^{A_o}\vec{\mathbf{r}}^{Q} \\
&= (1.5\,\widehat{\mathbf{a}}_x + 2.5\,\widehat{\mathbf{a}}_y + 1\,\widehat{\mathbf{c}}_x + {}^{-}2\,\widehat{\mathbf{c}}_y)\cdot(1.5\,\widehat{\mathbf{a}}_x + 2.5\,\widehat{\mathbf{a}}_y + 1\,\widehat{\mathbf{c}}_x + {}^{-}2\,\widehat{\mathbf{c}}_y) \\
&= 1.5 * 1.5\,(\widehat{\mathbf{a}}_x\cdot\widehat{\mathbf{a}}_x) + 1.5 * 2.5\,(\widehat{\mathbf{a}}_x\cdot\widehat{\mathbf{a}}_y) + 1.5 * 1\,(\widehat{\mathbf{a}}_x\cdot\widehat{\mathbf{c}}_x) + 1.5 * {}^{-}2\,(\widehat{\mathbf{a}}_x\cdot\widehat{\mathbf{c}}_y) \\
&\quad + 2.5 * 1.5\,(\widehat{\mathbf{a}}_y\cdot\widehat{\mathbf{a}}_x) + 2.5 * 2.5\,(\widehat{\mathbf{a}}_y\cdot\widehat{\mathbf{a}}_y) + 2.5 * 1\,(\widehat{\mathbf{a}}_y\cdot\widehat{\mathbf{c}}_x) + 2.5 * {}^{-}2\,(\widehat{\mathbf{a}}_y\cdot\widehat{\mathbf{c}}_y) \\
&\quad + 1 * 1.5\,(\widehat{\mathbf{c}}_x\cdot\widehat{\mathbf{a}}_x) + 1 * 2.5\,(\widehat{\mathbf{c}}_x\cdot\widehat{\mathbf{a}}_y) + 1 * 1\,(\widehat{\mathbf{c}}_x\cdot\widehat{\mathbf{c}}_x) + 1 * {}^{-}2\,(\widehat{\mathbf{c}}_x\cdot\widehat{\mathbf{c}}_y) \\
&\quad + {}^{-}2 * 1.5\,(\widehat{\mathbf{c}}_y\cdot\widehat{\mathbf{a}}_x) + {}^{-}2 * 2.5\,(\widehat{\mathbf{c}}_y\cdot\widehat{\mathbf{a}}_y) + {}^{-}2 * 1\,(\widehat{\mathbf{c}}_y\cdot\widehat{\mathbf{c}}_x) + {}^{-}2 * {}^{-}2\,(\widehat{\mathbf{c}}_y\cdot\widehat{\mathbf{c}}_y)
\end{aligned}$$

Here we are blocked since we don't know the dot products between basis vectors of A and C. But we **do** know all the basis vectors are unit length, so the only remaining term in the dot product is cosine of the angle between those vectors, which we find from the figure. It is convenient to arrange all the possible combinations in a table:

${}^{C}R^{A}$	$\widehat{\mathbf{a}}_x$	$\widehat{\mathbf{a}}_y$	$\widehat{\mathbf{a}}_z$
$\widehat{\mathbf{c}}_x$	$\cos(30°)$	$\cos(90° - 30°)$	$\cos(90°)$
$\widehat{\mathbf{c}}_y$	$\cos(90° + 30°)$	$\cos(30°)$	$\cos(90°)$
$\widehat{\mathbf{c}}_z$	$\cos(90°)$	$\cos(90°)$	$\cos(0°)$

=

${}^{C}R^{A}$	$\widehat{\mathbf{a}}_x$	$\widehat{\mathbf{a}}_y$	$\widehat{\mathbf{a}}_z$
$\widehat{\mathbf{c}}_x$	0.86603	0.5	0
$\widehat{\mathbf{c}}_y$	−0.5	0.86603	0
$\widehat{\mathbf{c}}_z$	0	0	1

Some things to notice about the table above:

- This is often called a *rotation table* because it relates two bases rotated with respect to one another. The symbol ${}^{C}R^{A}$ is usually just read left-to-right using the names of the bases, e.g. "C-R-A".
- Each entry **is** the dot product between the corresponding row and column labels. For example, $\widehat{\mathbf{a}}_x \cdot \widehat{\mathbf{c}}_y = -0.5$.
- These entries are sometimes called *direction cosines* since they are the cosine of the angle between unit-length vectors.
- To form a unit vector from a linear combination of the unit vectors in the other basis, look at the corresponding row or column. For example, $\widehat{\mathbf{a}}_y = 0.5\,\widehat{\mathbf{c}}_x + 0.866\,\widehat{\mathbf{c}}_y + 0\,\widehat{\mathbf{c}}_z$.

Armed with this information, we can now finish the previous calculation:

$$
\begin{aligned}
\left|{}^{A_o}\vec{\mathbf{r}}^{\,Q}\right|^2 = \;& 1.5 * 1.5\ (1) && + 1.5 * 2.5\ (0) && + 1.5 * 1\ (0.866) && + 1.5 * \text{-}2\ (\text{-}0.5) \\
& + 2.5 * 1.5\ (0) && + 2.5 * 2.5\ (1) && + 2.5 * 1\ (0.5) && + 2.5 * \text{-}2\ (0.866) \\
& + 1 * 1.5\ (0.866) && + 1 * 2.5\ (0.5) && + 1 * 1\ (1) && + 1 * \text{-}2\ (0) \\
& + \text{-}2 * 1.5\ (\text{-}0.5) && + \text{-}2 * 2.5\ (0.866) && + \text{-}2 * 1\ (0) && + \text{-}2 * \text{-}2\ (1) \\
=\;& 12.938 \\
\left|{}^{A_o}\vec{\mathbf{r}}^{\,Q}\right| =\;& \sqrt{12.94} \approx 3.60 \text{ meters}
\end{aligned}
$$

What would it take to apply our familiar "square root of sum of squares" formula to find distance? We'd need ${}^{A_o}\vec{\mathbf{r}}^{\,Q}$ to be expressed in terms of only basis A or C, but not both. How do we do this?

Recall the table above allows us to express a unit vector from one basis in terms of the other basis. Hence, we can simply substitute and combine terms:

$$
\begin{aligned}
{}^{A_o}\vec{\mathbf{r}}^{\,Q} &= 1.5\ \widehat{\mathbf{a}}_x + 2.5\ \widehat{\mathbf{a}}_y + 1\ \widehat{\mathbf{c}}_x + \text{-}2\ \widehat{\mathbf{c}}_y \\
&= 1.5\ \widehat{\mathbf{a}}_x + 2.5\ \widehat{\mathbf{a}}_y + 1\ (0.866\ \widehat{\mathbf{a}}_x + 0.5\ \widehat{\mathbf{a}}_y + 0\ \widehat{\mathbf{a}}_z) + \text{-}2\ (\text{-}0.5\ \widehat{\mathbf{a}}_x + 0.866\ \widehat{\mathbf{a}}_y + 0\ \widehat{\mathbf{a}}_z) \\
&= 3.37\ \widehat{\mathbf{a}}_x + 1.27\ \widehat{\mathbf{a}}_y
\end{aligned}
$$

$\left|{}^{A_o}\vec{\mathbf{r}}^{\,Q}\right| = \sqrt{(3.37^{\,2} + 1.27^{\,2})} \approx 3.60$ meters, which matches the previous result.

There are a variety of reasons one might pick a particular basis to work in, but the vector ${}^{A_o}\vec{\mathbf{r}}^{\,Q}$ has the **same magnitude** and **same direction** whether it is *expressed* in terms of basis A, basis C, or some combination of the two. When we change how a vector is expressed using a rotation table it's best to use the words "re-express" or perhaps "change basis". We avoid saying that the vector was "rotated" (or "converted", or "transformed"). because it's still the same vector!

3 Toward Rotation and Transformation Matrices

In the previous section we used a relatively verbose notation free of matrices or linear algebra, mostly to prove a point. In practice it is convenient to use matrices, but we have to be careful. Consider the following statement, which might happen if you naively type the given position vectors into a computer program:

$$
{}^{A_o}\vec{\mathbf{r}}^{\,Q} = {}^{A_o}\vec{\mathbf{r}}^{\,C_o} + {}^{C_o}\vec{\mathbf{r}}^{\,Q} \overset{?}{=} \begin{bmatrix} 1.5 \\ 2.5 \\ 0 \end{bmatrix} + \begin{bmatrix} 1 \\ \text{-}2 \\ 0 \end{bmatrix} \overset{\text{(wrong)}}{=} \begin{bmatrix} 2.5 \\ 0.5 \\ 0 \end{bmatrix} \overset{\text{(wrong)}}{=} 2.5\ \widehat{\mathbf{a}}_x + 0.5\ \widehat{\mathbf{a}}_y
$$

The problem is that matrices are meaningless collections of numbers. Matrices are convenient and powerful tools of modern computing, but you have to carry enough meaning and context in your notation that you can identify mistakes. For example, a subscript can denote that a matrix represents a linear combination of specific basis vectors:

$$
{}^{A_o}\vec{\mathbf{r}}^{\,Q} \overset{\text{(ok)}}{=} \begin{bmatrix} 1.5 \\ 2.5 \\ 0 \end{bmatrix}_{\widehat{\mathbf{a}}_{xyz}} + \begin{bmatrix} 1 \\ \text{-}2 \\ 0 \end{bmatrix}_{\widehat{\mathbf{c}}_{xyz}} \qquad \text{Can't simplify further!}
$$

In the previous section we were able to arrive at a single-basis expression by substituting $\widehat{\mathbf{c}}_x$, $\widehat{\mathbf{c}}_y$, $\widehat{\mathbf{c}}_z$ with linear combinations of $\widehat{\mathbf{a}}_x$, $\widehat{\mathbf{a}}_y$, $\widehat{\mathbf{a}}_z$ using the values in the table ${}^{C}R^{A}$.

Linear algebra is excellent for expressing linear combinations! Multiplying a 3x3 matrix with a 3x1 matrix gives a 3x1 matrix that is a linear combination of the columns of the 3x3 matrix. By treating the rotation table as a matrix then it's straight forward to re-express a vector by matrix multiplication. Let's not forget what's happening here – the columns of the rotation matrix **must** represent the same basis vectors as those in column matrix we wish to re-express, otherwise the meaning of the arithmetic has changed.

To take ${}^{C_o}\vec{\mathbf{r}}^{\,Q}$ which was given in basis C and re-express it in A, we need a matrix whose columns represent basis C. This is ${}^{A}R^{C}$, which is the transpose of the table we wrote before – it contains the same information but the rows and columns (including the unit vector labels) are swapped. Then we can do the following, which is exactly equivalent to the substituation we did before:

$$
{}^{A_o}\vec{\mathbf{r}}^{\,Q} \overset{(\text{ok})}{=} \begin{bmatrix} 1.5 \\ 2.5 \\ 0 \end{bmatrix}_{\hat{\mathbf{a}}_{xyz}} + \begin{bmatrix} 0.866 & -0.5 & 0 \\ 0.5 & 0.866 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ -2 \\ 0 \end{bmatrix}_{\hat{\mathbf{c}}_{xyz}}
$$

$$
= \begin{bmatrix} 1.5 \\ 2.5 \\ 0 \end{bmatrix}_{\hat{\mathbf{a}}_{xyz}} + \begin{bmatrix} 1.866 \\ -1.23 \\ 0 \end{bmatrix}_{\hat{\mathbf{a}}_{xyz}} = \begin{bmatrix} 3.366 \\ 1.27 \\ 0 \end{bmatrix}_{\hat{\mathbf{a}}_{xyz}} = 3.366\ \hat{\mathbf{a}}_x + 1.27\ \hat{\mathbf{a}}_y
$$

3.1 Coordinate Frames and "Transforming" Vectors

In many fields (e.g. graphics, robotics) a "rigid frame" or "coordinate frame" is the combination of a vector basis and a special point called the "origin" of the coordinate frame. In this example, "coordinate frame C" (often abbreviated "frame C") could have C_o as its origin and $\hat{\mathbf{c}}_x$, $\hat{\mathbf{c}}_y$, $\hat{\mathbf{c}}_z$ as its basis vectors.

By convention, vectors in these fields nearly always implicitly couple the "tail" of the vector and the basis used to express the vector. For example, by convention ${}^{C_o}\vec{\mathbf{r}}^{\,Q}$ will always be expressed in terms of C and might be called "the position of Q in (or relative to) frame C". The notation in code specifies only the coordinate frame since the tail point is implied: c_p_q. Similarly, ${}^{A_o}\vec{\mathbf{r}}^{\,Q}$ will always be expressed in A, will be called "the position of Q in frame A", and might be notated in code as a_p_q.

It should be obvious by now, but ${}^{A_o}\vec{\mathbf{r}}^{\,Q}$ and ${}^{C_o}\vec{\mathbf{r}}^{\,Q}$ are **different** vectors (different mangitude and direction)! To calculate ${}^{A_o}\vec{\mathbf{r}}^{\,Q}$, which by convention must be expressed in A, you take ${}^{C_o}\vec{\mathbf{r}}^{\,Q}$ and re-express it in A, and then add ${}^{A_o}\vec{\mathbf{r}}^{\,C_o}$ as we did above.

a_p_q = a_p_c + a_R_c ∗ c_p_q;

This process of re-expressing a vector and then adding an offset is often called "transforming" it from one frame to another. It's misleading terminology – it would be better to call it "getting a point's position relative to a different coordinate frame origin and expressed in that basis" but admittedly that is a mouthful.

Notice that transforming c_p_q to frame A uses two pieces of information: ${}^{A_o}\vec{\mathbf{r}}^{\,C_o}$ expressed in A (a_p_c) and ${}^{\mathrm{A}}R^{\mathrm{C}}$ (a_R_c). These two items are generally packed together into a transformation object ${}^{\mathrm{A}}T^{\mathrm{C}}$ (read "A-T-C") denoted as a_T_c in code, and used in code as:

a_p_q = a_T_c ∗ c_p_q;

This information is sometimes packed into a 4x4 transformation matrix so that the mathematical expression above is simply matrix multiplication. However, in programming there are many ways to create an object that carries the semantics of a transformation matrix without actually being a matrix, so I think it's better to continue with the abstraction and avoid getting stuck on any specific representation.

The code notation above is insufficient for vectors with tails not originating at the coordinate frame origin. For example, if ${}^{C_o}\vec{\mathbf{r}}^{\,Q}$ is re-expressed in A using a rotation matrix (as we did above), what should we call it? It's still the position of Q measured from C_o, but it's measured in basis A. So it's misleading to say "the position of Q in frame A" (because it's not a vector from A_o to Q) and misleading to say "the position of Q in frame C" (because it's not expressed in basis C). In this case the notation must explicitly state how the vector is expressed in addition to what it represents physically:

a_vec_c_q $= \left[{}^{C_o}\vec{\mathbf{r}}^{\,Q}\right]_{\hat{\mathbf{a}}_{xyz}}$

3.2 On Language

When referring to the rotation matrix ${}^{\mathrm{A}}R^{\mathrm{C}}$ in spoken English, people tend to use the semantics of input / output because of how matrix arithmetic works. For example, you might hear someone call ${}^{\mathrm{A}}R^{\mathrm{C}}$ the "rotation matrix from C to A" or in code call it a_from_c because of how it's **used** to take a vector expressed in C and **express** it in A. A similar thing happens with transformation matrices, where ${}^{\mathrm{A}}T^{\mathrm{C}}$ might be described as "the transformation from C to A".

But the symbol name is usually just read left-to-right using the letters ("A-R-C" or "A-T-C"). This mismatch between reading the symbol name and the right-to-left matrix operator description is completely an artifact of matrix operations. Just to prove the point: I could write a C++ function that takes ${}^{\mathrm{A}}R^{\mathrm{C}}$ and a vector in terms of $\hat{\mathbf{a}}_x$, $\hat{\mathbf{a}}_y$, $\hat{\mathbf{a}}_z$ and returns the same vector expressed in $\hat{\mathbf{c}}_x$, $\hat{\mathbf{c}}_y$, $\hat{\mathbf{c}}_z$ – suddenly it's "the rotation matrix from A to C" simply because my function isn't tied to linear algebra and matrix multiplication semantics!

So, my recommendation is to **avoid** the long-form description for matrices! Just read the symbol name left-to-right, know how it's used, and move on.