Quaternions White Paper

November 30, 2003

NAIF

This document discusses the application of quaternions for robotic spacecraft orientation encapsulation – also referred to as attitude or pointing. It is written to connect formal mathematical presentations of quaternion arithmetic with documents and software within NASA's SPICE ancillary information system. The primary intent of these notes is to eliminate future data system problems that might stem from the absence of standards associated with quaternions and the mathematical underpinnings.

The NAIF documents "Rotations Required Reading" and "C-kernel Required Reading" contain consistent specifications and examples for the use of quaternions in the SPICE context. These conventions carry through to subroutine modules provided in SPICELIB (and CSPICE) to deal with quaternions. There are at least two distinct conventions for quaternion usage that are common in the literature, but neither of these NAIF documents addresses the subtle differences between them. This document is designed to help bridge that gap and draw attention to these subtleties.

This text is primarily composed in layman's terminology and language. It is intended to provide SPICE users the necessary information to interface with external data or software that may not conform to the SPICE quaternion model. This set of notes as written is neither completely rigorous nor comprehensive. For "complete" details the reader is referred to the references.

Quick Start

A description of the SPICE standard and a procedure for determining whether external quaternions conform are provided in subsequent sections of the present document.

If the quaternions that require importation into the SPICE system do not agree with the SPICE standard, then in most cases the following transformation is all that is required:

$$Q_s = (q_4, -q_1, -q_2, -q_3)$$

where the original, non-SPICE quaternion is $(q_1, q_2, q_{3,} q_4)$ and Qs is the equivalent SPICE style quaternion. Comparing the rotation matrix produced from the converted quaternion in the SPICE context and the original quaternion in its context should validate this transformation. If the matrices are identical then no additional work aside from the above transformation is required. In the event that they disagree then the remainder of this document will help assess possible causes.

Rotational Pitfalls

Before delving into the issues that surround quaternions and their association with rotations, it is useful to briefly review rotations in general and some points of possible confusion. The term rotation matrix is used throughout this document and for all purposes can be considered synonymous with the terms C-matrix and direction cosine matrix.

All descriptions of rotations, whether explicit or implicit, prescribe an axis about which rotation occurs and an angle (including a sign) specifying the amount and direction of rotation. Attempts at defining rotations without the aid of a rigorous mathematical system are often plagued with uncertainties revolving around what one means by frame, angle, and axis. Further, most misunderstandings have precisely the same effect—yielding rotation in the improper direction. The composition of several misunderstandings can lead to correct results in one context, incorrect in another, and frustration in general.

While one is inclined to believe complete mathematical systems designed to describe rotations leave little room for ambiguity or confusion, it is often one's preconceived notions brought into conflict with the established system that are the source of confusion here. In either of the two cases, before interfacing can be accomplished successfully both parties must be certain they are speaking precisely the same language.

Positive Angles of Rotation

Perhaps the first potential area of confusion lies in the description of the positive angle of rotation. The left and right hand rules provide a convenient mechanism for enumerating the two possible definitions. Take the thumb of the appropriate hand and point it in the direction of the axis of rotation. Then the remaining fingers on that hand curl around the axis in the positive direction of rotation. The right hand rule is almost always employed to describe positive rotations. For the remainder of this document, all positive rotations are characterized by the right hand rule.

Coordinate vs. Vector Rotations

Another area of confusion that is closely connected to the definition of angle is whether one is attempting to describe a coordinate system (frame) rotation or rotation of a vector within a fixed coordinate system. Consider the following diagram: (The invisible rotation axis (k) points out of the page or screen from the intersection of the i and j axes.)

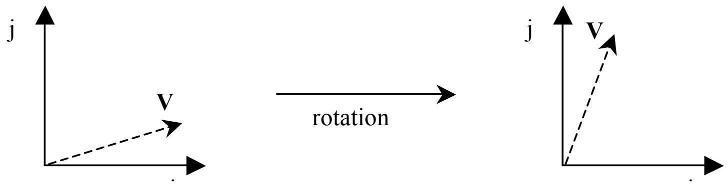

Fig. A: Vector Rotation

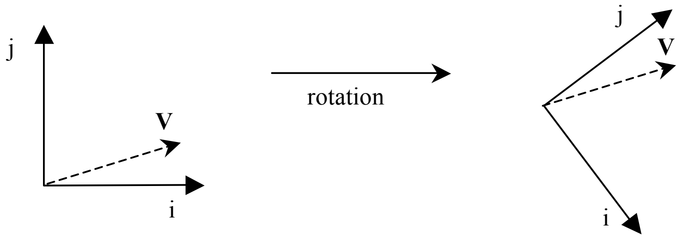

Fig. B: Coordinate System Rotation

The figure above illustrates two different views of precisely the same rotation. In **A** the vector **V** is rotated through a positive angle about the axis of rotation that points out of the page. In **B** the coordinate frame is rotated through an angle of the same magnitude that is negative in sign. Here lies the potential for confusion; when one says rotate a certain number of degrees about an axis, do they mean a vector rotation as in **A**, or a coordinate frame rotation as in **B**.

Matrix Multiplication: Left vs. Right

One of the most common means of encapsulating a rotation mathematically is through use of matrices. However usage of a matrix requires a common definition of the basic set of matrix operations especially multiplication. There are many possible ways one can effect matrix multiplication, but the following is standard fare in the mathematics community:

$$[V'] = \begin{bmatrix} & & \\ & M & \\ & & \end{bmatrix} [V]$$

Left matrix multiplication.

where V’ and V are the vectors and M is the matrix. However it is possible to define multiplication to occur on the right as shown below:

$$[V']^T = [V]^T \begin{bmatrix} & & \\ & M & \\ & & \end{bmatrix}$$

Right matrix multiplication.

Using the standard rules of matrix multiplication, taking a matrix intended for left multiplication, and applying it on the right produces a vector rotated about the same axis, but in the opposite direction.

Base and Target Frames

The last issue that this brief review of rotations addresses is the sense of rotation. Some refer to a rotation as from base frame to target frame, while others mean precisely the opposite. Confusion here may also result in a rotation about the proper axis but in the wrong direction.

Quaternions

A quaternion is a mathematical construct that consists of four individual numeric components. Quaternions are a convenient mechanism for encapsulating orientation information since they require only four units of numeric storage, as opposed to the nine needed for a rotation matrix. Unfortunately there are no standards defining construction of quaternions, the underlying associated mathematics, or the connection to rotations. In the absence of such standards, different organizations adopt disparate definitions, which makes communication difficult. It is the intent of the remainder of this document to draw the reader's attention to the two most commonly used quaternion systems in the robotic spacecraft community, and discuss the relevant issues in interpreting quaternions from other possible systems.

Let's begin by illuminating the commonalties across the quasi-standards in common use.

(1) One of the quaternion components is designated as the scalar, while the remaining three are typically referred to as the vector components. The reason for this is fairly straightforward; the vector components contain the information that specify the axis of rotation, while the scalar component is used to determine the magnitude of the rotation angle.

(2) Rotations obey the right hand rule.

(3) All quaternions that specify rotations are required to be of unit length. By this we simply mean the sum of the squares of the components is unity.

(4) Any Cartesian reference frame can be rotated into any other Cartesian frame using a single quaternion.

(5) There are two quaternions that yield the same end condition, but the rotation "path" taken is different.

Normally we're not interested in rotating one reference frame into another—rather we are interested in rotating (transforming) a vector specified in one frame into the other frame. Proper use of the quaternion does this. In the space science domain quaternions are frequently used to specify "pointing" of an instrument or a spacecraft. That lingo is a bit misleading: the quaternion is really used to specify the orientation of a Cartesian reference frame that is fixed to an instrument or to a spacecraft (or some spacecraft structure) relative to a "known" reference frame—the base frame. This specification of the orientation can then be utilized to develop the appropriate transformation.

As a point of interest, given a quaternion, one can construct an equivalent 3x3 rotation matrix and vice versa. However, formation of a rotation matrix is not required to use quaternions. Indeed, quaternions are used instead of matrices because they require less storage space, they avoid singularities, and they offer other useful properties.

SPICE Quaternions

The specifications used within the SPICE system—and thus as the PDS specification—are selected simply because the large collection of extant codes and documentation that follow this "standard," including the fact that the SPICE C-kernel, which is really a collection of quaternions (not direction cosine matrices), assumes this definition.

Let $Q = (q_0, q_1, q_2, q_3)$, be a unit quaternion.
Let ϕ be the magnitude of the coordinate system rotation angle from the base frame measured positively (using the right hand rule).
Let $R = (r_1, r_2, r_3)$ be a unit vector specifying the rotation axis, defined in the base frame.

In the SPICE standard, the 0th component is the scalar and the 1st, 2nd, and 3rd components constitute the vector components. Then the quaternion that encapsulates this rotation would be:

$$Q = \left(\cos\left(\frac{\phi}{2}\right), -r_1 \sin\left(\frac{\phi}{2}\right), -r_2 \sin\left(\frac{\phi}{2}\right), -r_3 \sin\left(\frac{\phi}{2}\right)\right).$$

A rotation matrix is formed from a quaternion constructed per the SPICE standard thusly:

$$M = \begin{bmatrix} 1-2\left(q_2^2+q_3^2\right) & 2\left(q_1q_2-q_0q_3\right) & 2\left(q_1q_3+q_0q_2\right) \\ 2\left(q_1q_2+q_0q_3\right) & 1-2\left(q_1^2+q_3^2\right) & 2\left(q_2q_3-q_0q_1\right) \\ 2\left(q_1q_3-q_0q_2\right) & 2\left(q_2q_3+q_0q_1\right) & 1-2\left(q_1^2+q_2^2\right) \end{bmatrix}.$$

One can effect rotations at the quaternion level. This is accomplished in the SPICE style quaternions through the following relations:

Let $A_b = (a_1, a_2, a_3)$ be a vector in the base frame.
Let $A_t = (t_1, t_2, t_3)$ be the coordinates of vector A_b in the target frame.
Let $Q = (q_0, q_1, q_2, q_3)$ be the SPICE style quaternion that describes the rotation that transforms A_b into A_t.

Define the conjugate of Q to be $Q^* = (q_0, -q_1, -q_2, -q_3)$. Note we are just negating the vector components of the quaternion.

Permit the extension of the three component A_b to a quaternion having four components as $A_b^q = (0, a_1, a_2, a_3)$. Note the choice of the scalar component as zero is arbitrary, since whatever value is selected is preserved by the rotation.

Define the product of two quaternions to be:

$$Y \bullet Z = \left(y_0 z_0 - \langle y_v, z_v \rangle\right) + \left(y_0 z_v + z_0 y_v + y_v \times z_v\right)$$

where y_0 and z_0 are the scalar components of Y and Z respectively, and y_v and z_v are the vector components. Note $\langle , \rangle$ denotes the standard scalar or inner product, and $\times$ the standard vector or cross product.

The quaternion Q can be utilized to compute the appropriate rotation:

$$A_t^q = Q \bullet A_b^q \bullet Q^*,$$

where A_t^q and A_b^q are the target and base frame vectors extended to quaternions as specified above.

Alternate Style Quaternions

The other popular quaternion specification is discussed in detail in this section. The relations will look nearly identical to the SPICE style discussed above; as such it is important to READ CAREFULLY and to clearly understand that all of the material in this section is part of the alternate specification. Contrasting comments and necessary operations for conversion will be held for the next section.

Let $Q = (q_1, q_2, q_3, q_4)$, be a unit quaternion.
Let ϕ be the magnitude of the coordinate system rotation angle from the base frame measured positively (using the right hand rule).
Let $R = (r_1, r_2, r_3)$ be a unit vector specifying the rotation axis, defined in the base frame.

In this style, the 4th component is the scalar and the 1st, 2nd, and 3rd components constitute the vector components. Then the quaternion that encapsulates this rotation would be:

$$Q = \left(r_1 \sin\left(\frac{\phi}{2}\right), r_2 \sin\left(\frac{\phi}{2}\right), r_3 \sin\left(\frac{\phi}{2}\right), \cos\left(\frac{\phi}{2}\right) \right).$$

And the associated rotation matrix when multiplied by vectors in the base frame from the left produces vectors in the target frame:

$$M = \begin{bmatrix} 1-2\left(q_2^2+q_3^2\right) & 2\left(q_1q_2+q_4q_3\right) & 2\left(q_1q_3-q_4q_2\right) \\ 2\left(q_1q_2-q_4q_3\right) & 1-2\left(q_1^2+q_3^2\right) & 2\left(q_2q_3+q_4q_1\right) \\ 2\left(q_1q_3+q_4q_2\right) & 2\left(q_2q_3-q_4q_1\right) & 1-2\left(q_1^2+q_2^2\right) \end{bmatrix}.$$

One can effect rotations at the quaternion level. This is accomplished using this alternate style quaternion through the following relations:

Let $A_b = (a_1, a_2, a_3)$ be a vector in the base frame.
Let $A_t = (t_1, t_2, t_3)$ be the coordinates of vector A_b in the target frame.
Let $Q = (q_1, q_2, q_3, q_4)$ be the alternate style quaternion that describes the rotation, that transforms A_b into A_t.

Define the conjugate of Q to be $Q^* = (-q_1, -q_2, -q_3, q_4)$. Note we are just negating the vector components of the quaternion.

Permit the extension of A_b to a quaternion as $A_b^q = (a_1, a_2, a_3, 0)$. The choice of the scalar component as zero is arbitrary.

Define the product of two quaternions to be:

$$Y \otimes Z = \left(y_4 z_4 - \langle y_v, z_v \rangle\right) + \left(y_4 z_v + z_4 y_v - y_v \times z_v\right)$$

where y_4 and z_4 are the scalar components of Y and Z respectively, and y_v and z_v are the vector components. Note $\langle , \rangle$ denotes the standard scalar or inner product, and $\times$ the standard vector or cross product.

The quaternion Q can be utilized to compute the appropriate rotation:

$$A_t^q = Q \otimes A_b^q \otimes Q^*,$$

where A_t^q and A_b^q are the target and base frame vectors extended to quaternions as specified above.

Example

Consider the following example:

Let $\phi = \arctan\left(\frac{4}{3}\right) \cong 53.13^o$ be the coordinate system rotation angle.
Let $R = (0,0,1)$ be the axis of the coordinate rotation.

SPICE Quaternions

The SPICE quaternion that represents this rotation is:

$$Q_S = \left(\frac{3}{5}, 0, 0, -\frac{4}{5}\right)$$

Which yields the following matrix when converted using the methodology discussed above:

$$M_S = \begin{bmatrix} -\frac{7}{25} & \frac{24}{25} & 0 \\ -\frac{24}{25} & -\frac{7}{25} & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

And now ask the question what coordinates

Alternate Style Quaternions

The alternate style quaternion that represents this rotation is:

$$Q_A = \left(0, 0, \frac{4}{5}, \frac{3}{5}\right)$$

Which yields the following matrix when converted using the methodology discussed above:

$$M_A = \begin{bmatrix} -\frac{7}{25} & \frac{24}{25} & 0 \\ -\frac{24}{25} & -\frac{7}{25} & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

And now ask the question what coordinates does the vector: $V_B = [1,1,0]^T$ have in the target frame.

$$V_T = \begin{bmatrix} -\frac{7}{25} & \frac{24}{25} & 0 \\ -\frac{24}{25} & -\frac{7}{25} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} \frac{17}{25} \\ -\frac{31}{25} \\ 0 \end{bmatrix}$$

However, one can elect to use quaternion arithmetic to effect this transformation. Extend V_B to a quaternion and apply the SPICE style multiplication to obtain:

$$V_T^q = Q_S \bullet V_B^q \bullet Q_S^*$$

$$V_B^q \bullet Q_S^* = \left(0, \frac{7}{5}, -\frac{1}{5}, 0\right)$$

$$V_T^q = Q_S \bullet \left(V_B^q \bullet Q_S^*\right) = \left(0, \frac{17}{25}, -\frac{31}{25}, 0\right)$$

does the vector: $V_B = [1,1,0]^T$ have in the target frame.

$$V_T = \begin{bmatrix} -\frac{7}{25} & \frac{24}{25} & 0 \\ -\frac{24}{25} & -\frac{7}{25} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} \frac{17}{25} \\ -\frac{31}{25} \\ 0 \end{bmatrix}$$

However, one can elect to use quaternion arithmetic to effect this transformation. Extend V_B to a quaternion and apply the alternate style multiplication to obtain:

$$V_T^q = Q_S \bullet V_B^q \bullet Q_S^*$$

$$V_B^q \bullet Q_S^* = \left(\frac{7}{5}, -\frac{1}{5}, 0, 0\right)$$

$$V_T^q = Q_S \bullet \left(V_B^q \bullet Q_S^*\right) = \left(\frac{17}{25}, -\frac{31}{25}, 0, 0\right)$$

So the end results are the same, if one ignores the permutation of scalar and vector components that occurs between the two systems.

Differences between Quaternion Specifications

There are two critical differences between the SPICE specification and the alternate presented above. The first and perhaps most obvious is the permutation of the scalar and vector components. The SPICE quaternions place the scalar first, while the alternate system reserves this for the last component. The less obvious and most impacting difference is the formulation of the multiplication operation. In the SPICE system the sign of the cross product component is positive, while in the alternate system this is negative. This difference has the effect of inverting the sense of the rotation, so to convert alternate style quaternions to the SPICE specification one need only define: $Q_s = (q_4, -q_1, -q_2, -q_3)$, where q_i are the original components as listed in the alternate specification.

Checklist

When importing quaternions from an external source to the SPICE system the following issues require examination:

(1) Examine the order of the external source quaternion components. Clearly identify the scalar and vector components. Identifying the order of the vector components is as important as recognizing them.

(2) Check the multiplication formulation. Make certain that the definition of multiplication is consistent with the SPICE definition outlined above. One may elect to examine the external source quaternion conversion to rotation matrices using the SPICE methodology, and if it agrees with the external source procedure for producing rotation matrices this is sufficient.

(3) Verify that the rotational issues discussed at the beginning of this document are clearly resolved. Namely, does the external source define quaternions that produce rotations from base frame to target frame as in SPICE, or vice versa. If necessary, check that the rotation matrix output by the external source is intended for left matrix multiplication. Lastly, check that the right hand rule is in effect.

(4) The final issue that one should be aware of when dealing with quaternions is that some systems may effect rotations through quaternion multiplication in the following way:

$$A_t^q = Q^* \bullet A_b^q \bullet Q$$

This is the same as:

$$A_t^q = \left(Q \bullet A_b^q \bullet Q^*\right)^*$$

So the use of quaternions from such a system often results in rotations occurring in the incorrect direction, since the conjugate effectively changes the sign of the rotation angle.

References

(1) Methods of Mathematical Physics – Volume I, by R. Courant and D. Hilbert. (See the discussion of spherical harmonics).

(2) The Theory of Spinors, by Elie Cartan.

(3) Elements of Abstract and Linear Algebra, by H. Paley and P. M. Weichsel.

(4) Algebra, by T. W. Hungerford.

(5) Geometry of Manifolds, by R. L. Bishop and R. J. Crittenden.

(6) Gravitation, by C. W. Misner, K. S. Thorne, and J. A. Wheeler. (See Chapter 41 – Spinors)

(7) “C-Kernel Required Reading”, NAIF Toolkit Documentation.

(8) “Rotations Required Reading”, NAIF Toolkit Documentation.