

Hexpentaquaternions: a two-hand quaternion algebra

Source: <http://sci.tech-archive.net/Archive/sci.math/2006-01/msg04863.html>

- *From:* "Peter M Jack" <math@xxxxxxxxxxxxxxxxxxxxxxxxxxxxx>
 - *Date:* Sun, 29 Jan 2006 20:59:27 -0500
-

A new method for solving quaternion equations.

Suppose we're given a quaternion equation

$$A[1]qB[1] + A[2]qB[2] + \dots + A[n]qB[n] = C$$

and required to solve for q ,

where $A[k], B[k], C, q$, are all quaternions from Hamilton's right-hand system $ij = +k$. So,

$$\begin{aligned} q &= q_0.1 + q_1.i + q_2.j + q_3.k \\ C &= c_0.1 + c_1.i + c_2.j + c_3.k \\ A[k] &= a[k,0].1 + a[k,1].i + a[k,2].j + a[k,3].k \\ B[k] &= b[k,0].1 + b[k,1].i + b[k,2].j + b[k,3].k \end{aligned}$$

and the right-hand basis elements obey the rules

$$\begin{aligned} i^2 &= j^2 = k^2 = -1, \\ ij &= -ji = +k \\ jk &= -kj = +i \\ ki &= -ik = +j \end{aligned}$$

To solve the equation, all we need to do is convert the B parameters into "left-hand" quaternions B' so that we can move them over to the other side of the q parameter, and write

$$A[1]B'[1]q + A[2]B'[2]q + \dots + A[n]B'[n]q = C$$

then we can aggregate the factors

$$(A[1]B'[1] + A[2]B'[2] + \dots + A[n]B'[n]) \cdot q = C$$

and invert the equation

$$q = (A[1]B'[1] + A[2]B'[2] + \dots + A[n]B'[n])^{-1} \cdot C$$

Hexpentaquaternions: a two-hand quaternion algebra

that's it.

Details can be found in my new paper,

"Hexpentaquaternions: a two-hand quaternion algebra."
reference # hypcx-20060129a

available online here,

<http://www.hypercomplex.com/research/emgrav/abs20060129a.html>

The .pdf, .dvi, .djvu versions of the paper have active url links to references around the web, but in the .ps paper these links are inactive.

enjoy

pmj

-
- Prev by Date: *Re: Contradicrtion-free mathemattics (The new nonstandard analysis*
 - Next by Date: *Re: Contradicrtion-free mathemattics (The new nonstandard analysis*
 - Previous by thread: *What constitutes mathematical maturity?*
 - Next by thread: *Derivate OFa number*
 - Index(es):
 - ◆ *Date*
 - ◆ *Thread*