Available online at www.joac.info

ISSN: 2278-1862

Journal of Applicable Chemistry

2016, 5 (1): 1-20

(International Peer Reviewed Journal)

Intelligent Pedagogical Reviews (IPR)

HotIce: Hands-on-tutorial for intelligent computational Evolution Part 1: Quaternions in Omnimetrics (QuO)

K. Ramakrishna[1] and R. Sambasiva Rao[2*]

1. Department of Chemistry, Gitam Institute of Science, Gitam University, Visakhapatnam, 530 017, **INDIA**
2. School of Chemistry, Andhra University, Visakhapatnam 530 003, **INDIA**

Email: karipeddirk@gmail.com, rsr.chem@gmail.com

Accepted on 20th January 2016

(Dedicated with profound respects to Dr. J.R. Isaac, former professor of computer science, IIT, Bombay on his eighty fifth birth anniversary)

ABSTRACT

Quaternions, invented by Hamilton 1863, are extended complex numbers. A quaternion is represented by quadruple of 1 and zeros and geometrically by four axes. It surmounted the classical gimbal lock. Also, smooth rotations of 3D-objects are possible, thus circumventing hurdle of classical methods where 3D-highways lead to dead ends. Quaternion frame not only solves all earlier tasks with vector analysis, but instrumental to successfully arrive at an answer for unsolved riddles in 3D-object rotations. Rotation through 720^o is achievable and finds extensive applications in proteins, missiles, space shuttles etc. Altogether, a new branch of computational mathematics grew and showed a new path in solving Schrodinger wave equation, optimizations, and nature inspired algorithms. It is a coveted tool in the armor of 21st century computational science.

Keywords: Quaternion, Complex (imaginary), Protein molecules, rotation, Schrodinger wave equation, optimization.

Contents

Quaternions in Omnimetrics (QuO)

INTRODUCTION

Backdrop: The complex numbers could be treated as a pair of real numbers and perform algebraic operations like addition and multiplication. With inspiration of reflection and relation between complex numbers and 2D-geometry, Hamilton had brain storming effort for many years to invent higher or bigger algebra which would revolutionize 3D-geometry. For instance, elements of real as well as complex numbers can be inverted (chart 1). But, elements of three dimensions $\left(xt = R^3\right)$ cannot be inverted. Further, for many years he had a stigma of multiplication of triples, although he could add and subtract them.

Chart 1: Inversion of elements of real and complex numbers

$\left(xr * \frac{1}{xr} = 1\right)$ Eqn.1	xr:	xreal	2.0	R^1
$\left(\frac{xc * conj(xc)}{\lvert xc \rvert^2} = 1\right)$ Eqn.2	xc	Complex number	1.0 + 3*i	R^2
		Conj(xc): Conjugate of xc		

Research tutorials: The prime focus of these research pedagogic bits is towards non-mathematics majors to comprehend state-of-knowledge mathematical procedures for application in disciplines of their expertise. The simple as possible (SAP) data analysis (vide infra) modules do not require even paper and pencil leave aside electronic gadgets. At the same time Matlab programs (not optimized to keep transparency of logic and one to one correspondence between formulae and implementation) enable one to solve even complicated tasks. Traversing from real → complex → quaternion → Octonion → …. and vice versa both in sequential and with multiple jumps develops an integrated picture of number systems for task on need basis. It is definitely not for a feel of advanced tech bits to hype up (false) scores of methodology. Further, many advanced features like expert system approach, (numerical, symbolic, method) knowledge bases inadvertently give first hand exposure to basic tools of 21st century metrics/knowledge extraction. This series of tutorials lay local paths (of how to use high tech tools for trivial small scale queries) merging into super highways in solving real time lifesaving mega inter-/intra-disciplinary tasks. In this communication, the recent applications [1-56] of quaternions in diverse disciplines and basics of quaternion algebra implementation in Matlab software are described.

Invention of Quaternion

On 16th of October, 1843, Hamilton was walking with his wife along the Royal Canal for a meeting of the Royal Irish Academy in Dublin. Momentously, it stuck to him that no 3D-normed division algebra exists and thus 4D-algebra was a necessity. Hamilton felt the galvanic circuit of thought suddenly closed (which was open till that moment) emanating the sparkles $\left(i^2 = j^2 = k^2 = ijk = -1\right)$. He cut (carved) these fundamental axioms on the stone of the Broome Bridge (Fig. 1). During rest of his life, he developed quaternion algebra and applications to geometry using exactly the same equations [57-60].

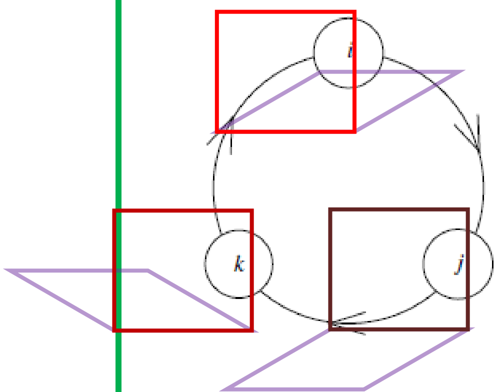


Fig.1: Quaternion plaque on Brougham (Broom) Bridge,Dublin

Here as he walked by
on the 16th of October 1843
Sir William Rowan Hamilton
in a flash of genius discovered
the fundamental formula for
quaternion multiplication
$i^2 = j^2 = k^2 = ijk = -1$
& cut it on a stone of this bridge

Chart 2: Categories of Algebras and oximes of quanternion

a	+b*i	+c*j +d*k
Real		
	Imaginary	
	Quaternion	

- q = a + b*i +c*j +d *k
- $xq=[a,\ V\{b,c,d\}]$
- Clifford algebra Frame, quaternions are represented as Cℓ0,2(R) ≅ Cℓ03,0(R).

a,b,c,d	Real numbers

Axioms of Quaternion

- Define basis elements $1 and (i, j, k)$
- Define quaternion as a vector space over reals $a+i*b+j*c+k*d$
- Define $(i^2 = j^2 = k^2 = -1)$

$\downarrow$

i^2 = -1	i*j = k	j*i = -k
j^2 = -1	j*k = i	k*j = -i
k^2 = -1	k*i = j	i*k = -j
i*j*k = -1		

i,j,k	Imaginary numbers
H	Hamilton
Q	Quaternion
a	Scalar part
V	Vector part

Axioms of Complex number

- Define basis elements — 1 and i
- Define complex number as a vector space over reals — a + i*b
- Define — $(i^2 = -1)$

- Quotient of two directed lines in a 3D-
- Quotient of two vectors
- 4D-associative normed division algebra over the real number domain

Classes of algebra s	
$$$ algebra	**$$$ algebra**
Clifford-	Banach-
Non-associative-	Bi-
Operator-	Diagram-
Quaternions-	Hopf-

2.1 Structure of quaternion: The quaternion is represented as quadruple containing four real numbers (a, b, c and d); the later three (b, c, d) in complex 3D-surface (Table 1, Fig.2). Thus, it is natural to conceive that quaternion -- miraculous frame for 3D-object rotations-- is an extension of complex numbers applicable to 2D-space. Geometrically x axis corresponds to pure real number and the rest of three axes to i, j, k to imaginary axes. If the real number is zero the quaternion becomes q = b*i +c*j +d *k and routine 3D-axes denotes complex surface, of course with special hidden characteristics. Thus, Q can also be deemed as axis (b, c, d) and angle (a) combination. The functioning of quaternions internally is very complicated and this is only superficial analogy to appreciate and not to probe into (illusionary) inferences. It is the first Non-associative algebra (chart 2) and Hamilton's quaternions are to $\mathbf{R}^4$ what complex numbers are to $\mathbf{R}^2$ space.

The first term, a, is thus often referred to as the scalar or sometimes the real part of the quaternion, and the triplet (b, c, d) as the vector part.

Table 1: Basis vector representation of quaternion, complex and real numbers

Basis	Quaternion	Complex		Real		
	q = a + b*i +c*j +d *k	Cmplx = a + b*i		Re = a		
	4D-	2D-	Quaternion domain 4D-	Real domain (1D- or R^1)	Complex domain 2D-	Quaternion domain 4D-
1	**(1 0 0 0)**	**(1 0)**	**(1 0 0 0)**	**(1)**	**(1 0)**	**(1 0 0 0)**
i	(0 1 0 0)	(0 1)	(0 1 0 0)		(0 0)	(0 0 0 0)
j	(0 0 1 0)		(0 0 0 0)			(0 0 0 0)

k	(0 0 0 1)		(0 0 0 0)			(0 0 0 0)
	One Real axis Three imaginary axes	One Real axis One imaginary axis		One Real axis		

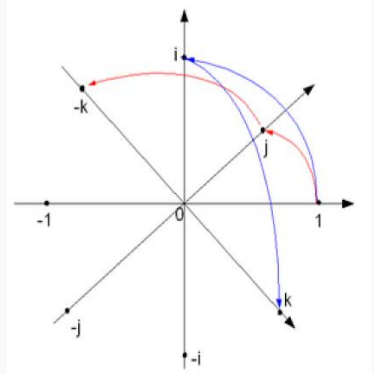


Fig. 2: Graphical representation of quaternion units product as 90°-rotation in 4D-space $ij = k, ji = -k, ij = -ji$

3. Applications

Quaternions find extensive applications in mathematical as well as in applied sciences of all disciplines. Initially, the focus was in theoretical and applied mathematics, in particular for calculations involving rotations in three-dimensions.

In quantum mechanics, intrinsic spin is represented with a maximum of 720 degrees and this is possible in only quaternion domain. But, the geometric rotations are limited to 360^{o}; then next cycle which is indistinguishable to the former continues. Einstein showed that gravity is the result of mass bending the local space-time domain. He also found the use of four dimensions to explain the constant speed of light for all inertial observers by invoking space time continuum or unification. Quaternions, first member of hyper complex frame probe into understanding quantum mechanical forces. The Pauli spin matrices and Maxwell equations are redefined in terms of quaternions. MRI (Magnetic Resonance Images) and CAT (Computed Axial Tomography) scans human/animal body masses for diagnostic purposes. The post processing performs rigid transformations represented by quaternions.

On line searches in Science Direct and ACS show all-pervading utility of quaternions in classical and modern research disciplines (Chart 3) ranging from solution of equations to quantum mechanics, molecular dynamics, modern physics, chemical-biology and physical chemistry-chemical physics. Typical research results in electrodynamics, NMR, proteins, quaternionic quantum theory, molecular modeling, conformation analysis, developing differential operators, transform techniques like FT, wavelet, mathematical solution methods (SVD, LS), Clifford algebra, Eigen values, coneigenvalues and Kalman filter are tabulated (Table 2).

Table 2: Typical research publications of interdisciplinary research in quaternion domain

Proteins	Ref
Global protein structure	26
Implicit Solvation	43
RNA sequences and tertiary structures.	9
Secondary structure	15
Helix parameters	15
Multiple Protein Structural Motif	24

Quaternionic quantum Mechanics	Ref
Nonrelativistic and relativistic molecular Kohn–Sham problem symmetry treatment and complex Least Squares coneigen-problem	14
Schrödinger equations	36
Quaternion LU decomposition	17

Rotation in Molecular Docking	18

Molecular clusters	Infinitesimal rotation	48
Molecular crystals	Molecular orientation for Use in latiice dynamics	55
NMR	Protein helices	53
NMR	Two-dimensional Fourier transform	41

Least squares	35,39
LU decomposition	8
Q-LSQR	30
Total least squares problem	1
Wavefunctions	56

Pearson type II-Riesz distribution	9
Molecular Conformation Sets	25
Molecular modeling	40,45

RF excitation waveforms First and second order derivatives	10
Solutions quaternion matrix equations XF # AX = BY	23
Spin-1 excitation	50
Surface coils	
SVD	13
Transition-state search	3
Wavelet transform	27
H-Hermicity,real quaternion matrix equations	5

Chart 3: Quaternions Application domains

Mathematics	Abstract algebra
Physics	Relativity Quantum mechanics
Computer science	3D-Graphics Animation
Chemical biology	Proteins
Aeronautics	satellites space shuttles
Medical diagnosis	

Initiation: Unit quaternions are initiated in om_unitQuat.m and the output is in ML_Fn 01.

ML_Fn 01: initiation of unit quaternion

```
unitQuatXYZ =
   0.7071   0.7071        0        0
   0.7071        0   0.7071        0
   0.7071        0        0   0.7071
```

```
%
% om_unitQuat.m (R S Rao 10/11/15)
%
function [unitQuatXYZ] = om_unitQuat
sc = 0.7071;
unitQuatXYZ = [ones(3,1), eye(3)] *sc
```

4.0 Operations on quaternions

The quaternion (xq) is also represented as [a,V], where V = (b,c,d), the complex part. If a = 0, then xq= [V(a,b,c)].

ॐ Conjugate [Conj(xq)]:

The conjugate of a quaternion is denoted as xq* analogous to complex number domain, or as conj(xq) adhering to algorithmic style. It is calculated by scalar multiplication of V by minus one, keeping the real scalar as it is.

The implementation of Formula.1 in Matlab function (om_conjugateQuat.m) and SAP data are given in ML_Fn 02. The program AutoTest_conj.m outputs results for typical test cases with and without real scalar term.

ML_Fn 02: Conjugate of quaternion [Conj(q)]:

$$q = a + b*i + c*j + d*k$$

$$conj(q) = a - b*i - c*j - d*k \qquad Formula.1$$

$$or\ conj(q) = \left[a,\ -b*i - c*j - d*k(-1)*\left[b,c,d\right]\ \right]$$

$$conj(conj(q)) = a + b*i + c*j + d*k \equiv q \quad Formula.2$$

```
om_conjugateQuat
xQuat =
  -2.6458   1.4142  -1.7321   2.0000
xQuatConjugate =
  -2.6458  -1.4142   1.7321  -2.0000
```

```
%
%om_conjugateQuat.m   (R S Rao) 12-11-15 ;
%
function [xQuatConjugate ]= om_conjugateQuat(xQuat)
%
if nargin ==0
  xQuat = [-sqrt(7),sqrt(2),-sqrt(3),2]
end
%
xr = xQuat(:,1:1);
xc = -1 * xQuat(:,2:4);
xQuatConjugate =  [xr,xc]
```

Properties of conjugate of a quaternion: The numerical expert system mode execution is in KB segment of Prop_conjQ.m. One can probe into each step of this algorithm without any computing or programming paradigm also.

```
            Properties Conjugate of quaternion
         ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
xQ =
   1   2   3   4
conjxQ =
   1  -2  -3  -4
conj_conjxQ =
   1   2   3   4
       conj{conj(xQ)} is equal to  Q since,(xQ - conj_conjxQ)==0

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
~~~
            Quaternion with zero real term
            ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
xQ =
   0   2   3   4
conjxQ =
   0  -2  -3  -4
conj_conjxQ =
   0   2   3   4
```

```
%
% AutoTest_conj.m
%
function AutoTest_conj
clean
dispst('Properties Conjugate of
quaternion') %
Prop_conjQ
dispst('Quaternion with zero real
term');
xQ = [0 2 3 4]
Prop_conjQ(xQ)
```

```
%
% Prop_conjQ.m     (R S Rao)
10/11/15; 10-7-05
%
function   Prop_conjQ(xQ)
if nargin ==0
   xQ = [1 2 3 4]
end
```

conj{conj(xQ)} is equal to Q since,(xQ - conj_conjxQ)==0	`[conjxQ]= om_conjugateQuat(xQ)`
	`[conj_conjxQ]=om_conjugateQuat(conjxQ)`
	`%`
	`% Numerical knowledge based inference`
	`Ant{:,1} = '(xQ - conj_conjxQ)==0';`
	`Conseq{:,1} = 'conj{conj(xQ)} is equal to Q';`
	`since = ' since,';`
	`if (eval(Ant{:,1}))`
	`dispst([Conseq{:,1}, since, Ant{:,1}])`
	`end`

ॐ Norm of quaternion [EuclNorm(xq)]

The product of a quaternion and its conjugate is $(a^2+b^2+c^2+d^2)$. In real number domain, the result is the square of a Euclidean length. Considering triplet $[b,c,d]$ as a vector $V\left(=[b,c,d]\right)$, the product is $q*conjq=\left(a^2+\|V\|^2\right)=\left(a^2+\|[b,c,d]\|^2\right)$. The length of quaternion is equal to square root of product quaternion and its conjugate (ML_Fn03). The norm or length of quaternion is also called modulus or quaternion sign (sign(q)).

$q*conjq=(a+b*i+c*j+d*k)*(a-b*i-c*j-d*k)$ $=(a^2-b^2*i^2-c^2*j^2-d^2*k^2)+(-a*b+b*a-c*d+d*c)*i+(-a*c+b*d+c*a-d*b)*j+(-a*d+b*c-c*b+d*a)*k$ $=(a^2+b^2+c^2+d^2)$	$xq^T=[a\quad b\quad c\quad d]^T=\begin{bmatrix}a\\b\\c\\d\end{bmatrix}$ $xq*xq^T=[a\quad b\quad c\quad d]*\begin{bmatrix}a\\b\\c\\d\end{bmatrix}$ $=(a^2+b^2+c^2+d^2)\quad Formula.3$

$$norm(xq)=\sqrt[2]{xq*conjxq}=\sqrt[2]{(a^2+b^2+c^2+d^2)}=\sqrt[2]{xq*xq^T}\qquad Formula.\ 4$$

ML_Fn 03: Norm of quaternion	

$\lvert q\rvert = q*conjq$
om_normQuat;
xQuat =
	`%`
	`% om_normQuat.m (R S Rao 10/11/15)`
	`%`
	`function [norm_xQuat]= om_normQuat(xQuat)`
	`if nargin ==0`

```
  sqrt(7)   2.0000   3.0000   4.0000
norm_xQuat =
   6
```

```
    xQuat = [sqrt(7),2,3,4]
end
    sqx = (xQuat * xQuat') ;
    norm_xQuat = sqrt(xQuat * xQuat') ;  %Formula. 4
```

```
[normalizedQuatNumber]= quat2normquat;
xQuat =
   2.6458   2.0000   3.0000   4.0000
norm_xQuat =
   6
normalizedQuatNumber =
   0.4410   0.3333   0.5000   0.6667
```

Table 3:Quaternions --> normalized quaternions

	Xq	Norm(xq)	xqnormalized
C	[4 3 0 0]	5	[0.8000 0.6000]
R	[6 0 0 0]	6	1
C	0.7071 0.7071 0 0	1.00	0.7071 0.7071 0 0

R	[0 0 0 0]	0	Caution: Quaternary number is zero! Norm does not exist as Square root of zero is not valid

```
%
% om_quat2normquat.m    (R S Rao
10/11/15)
%
function [normalizedQuatNumber]=
Om_quat2normquat(xQuat)
if nargin ==0
   xQuat = [sqrt(7),2,3,4]
end
[norm_xQuat]= om_normQuat(xQuat)

if norm_xQuat ~= 0
   normalizedQuatNumber =
xQuat/norm_xQuat
else
    caution{1,:}='Caution: Quaternary
number is zero!';
    caution{2,:}=' Norm does not exist';
    reason      = 'Square root of zero
is not valid';
    disp( [caution{1,:}, caution{2,:},
...
        ' as ', reason])
    normalizedQuatNumber = []
    return
end
```

ॐ Inverse of quaternion

Every quaternion except zero has an inverse. It is represented with a superscript of -1 or symbolically as inv(xq). The algorithm runs through calculation of conjugate and length of quaternion (alg.01). The application is in division operation of two quaternions. ML_Fn 04 describes output of om_inverseQuat.m for default quaternion (xq = [-sqrt(7),sqrt(2),-sqrt(3),2]).

Alg. 01: Inverse of quaternion

Input: xq ([a, b, c, d]

$$Step\ 1:\ Cal\ conjugate\ of\ xq \rightarrow conjXq$$

$$Step\ 2:\ Cal\ norm\ of\ xq \rightarrow normXq$$

$$Step\ 3:\ inv\_xq = \frac{conjXq}{(normXq)^2} \quad Formula.5$$

```
%
%  om_inverseQuat.m  (R S Rao 12-11-15)
%
function [invXq]= om_inverseQuat(xq)

if nargin ==0
   xq = [-sqrt(7),sqrt(2),-sqrt(3),2]
end
%%
    omcalled(' om_Conjugate ')
[conjXq]= om_conjugateQuat(xq) %Step 1
    omexit(' om_conjugate ')
    omcalled(' om_normQuate ')
[normXq]    = om_normQuat(xq)  % step 2
    omexit(' om_normQuate ')
invXq     =  [conjXq]./([normXq].^2);%
Formula 5
%%
```

ML_Fn 04: Output of Inverse of quaternion

```
>> om_inverseQuat
xQuat =
  -2.6458    1.4142   -1.7321    2.0000

      Calling  om_Conjugate .m
xQuatConjugate =
  -2.6458   -1.4142    1.7321   -2.0000
       exit from  om_conjugate .m

     Calling  om_normQuate .m

norm_xQuat =
     4
      exit from  om_normQuate .m

ans =
  -0.1654   -0.0884    0.1083   -0.1250
```

Table format

Table 4a: Inverse (Quaternion)

q	[-sqrt(7),sqrt(2),-sqrt(3),2 -2.6458 1.4142 -1.7321 2.0000
Conjugate(q)	-2.6458 -1.4142 1.7321 -2.0000
Norm(q)	4.0
Sqaure of Norm(q)	16
Inv(q) i.e. invq	-0.1654 -0.0884 0.1083 -0.1250

Table 4b

Inverse (inv(q))	
invq	-0.1654 -0.0884 0.1083 -0.1250
Conjugate(invq)	-0.1654 0.0884 -0.1083 0.1250
Norm(invq)	0.2501
Square of Norm(invq)	0.0626
Inv(invq) i.e. q	-2.6453 1.4138 -1.7321 1.9992

Table 4c

[xQuatInverse]= om_inverseQuat([1 0 1 0])

Inverse (Quaternion)	
q	1 0 -1 0
Conjugate(q)	1 0 -1 0
Norm(q)	4
Square of Norm(q)	16
Inv(q)	0.2500 0 0 0

Table 4d

[xQuatInverse]= om_inverseQuat([1/4 0 0 0])	
Inverse (scalarReal)	
q	0.2500 0 0 0
Conjugate(q)	0.2500 0 0 0
Norm(q)	0.25
Sqaure of Norm(q)	0.0625
Inv(q)	4
Inv(inv(q))	0.25
diff	0 0 0 0
inv{inv(xQ)} is equal to Q since,(abs(inv_invxq-xq))< eps ~~~~~~ eps: 2.2204e-16	

Table 4e

[xQuatInverse]= om_inverseQuat([2 3 0 0])	
Inverse (complexNumber)	
q	2 3 0 0
Conjugate(q)	2 -3 0 0
Norm(q)	3.6056
Sqaure of Norm(q)	13.0004
Inv(q)	0.1538 -0.2308 0 0

Inv(inv(q))	2.0000 3.0000 0 0
diff	1.0e-15 * 0.6661 0.8882 0 0
inv{inv(xQ)} is equal to Q since,(abs(inv_invxq-xq))< 1e-10 ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~	

Auto test program for inverse of quaternion: Here, a proper quaternion, complex number and scalar real values are tested with om_inverseQuat.m matlab function. The program Prop_inv.m is developed to demonstrate the equivalence of (xq^{-1}) *(xq) and (xq * xq^{-1}) and each product is equal to [1 0 0 0]. The second property is inverse of inverse (xq) is xq itself. The absolute difference is less than most of computational accuracies (10^{-10} or eps: 2.2204e-16). Table 4b shows the results of inverse of inverse_(xq) calculated in table 4a. The inverse of scalar real value of (1/4) (Table 4d) and complex number ([2 - 3*i]) (Table 4e) are obtained by quaternion inverse procedure. These calculations can also be practised with ease without any gadgets.

```
%
% Prop_inv.m    (R S Rao 10/11/15)
%
function   Prop_invQ(xq)
if nargin ==0
   xq = [sqrt(7),2,3,4]

end
invxq =  om_inverseQuat(xq);
dispst('inv(xq)');invxq
dispst('xq * invxq');
[xqMulinvxq]= om_prodQuat(xq,invxq)
dispst('invxq * xq')
[invxqMulxq]= om_prodQuat(invxq,xq)
 %
 dispst('inv(invxq)')
inv_invxq =  om_inverseQuat(invxq)
 diff=abs(inv_invxq-xq);diff
  since = ' since,';
 %
 Ant{:,1} = '(abs(inv_invxq-xq))< 1e-10';
 Conseq{:,1} = 'inv{inv(xq)} is equal to  xq';
 %
 Ant{:,2} = '(abs(xqMulinvxq-invxqMulxq ))< 1e-10';
 Conseq{:,2} = '  xq * invxq is equal to invxq *
xq';
 %
 Ant{:,3} = 'abs(xqMulinvxq - [1 0 0 0])< 1e-10';
 Conseq{:,3} = 'xq * invxq is equal to  [1 0 0 0]';
 %
 Ant{:,4} = 'abs(invxqMulxq - [1 0 0 0])< 1e-10';
```

```
%
% AutoTest_inv.m   (R S Rao)
%
function AutoTest_inv
clean
dispst('inverse of quaternion') %
xq = [-sqrt(7),sqrt(2),-sqrt(3),2]
[invxq]=
om_inverseQuat(xq),Prop_inv(xq)
dispst('inverse of Q with zero R') %
xq = [ 0, 2 3 4];
[invxq]=
om_inverseQuat(xq),Prop_inv(xq)

dispst('inverse of Real') %
xq = [ 2, 0 0 0];
[invxq]=
om_inverseQuat(xq),Prop_inv(xq)
```

```
Conseq{:,4} = 'xq * invxq is equal to  [1 0 0 0]';
%
%
for ii = 1:4
    if (eval(Ant{:,ii}))
        dispst([Conseq{:,ii}, since, Ant{:,ii}])
    end
end
disp(' ')
xq,invxq, inv_invxq ,xqMulinvxq, invxqMulxq
```

Table 05: Auto test cases for inverse of quaternion

	Quaternion	q with zero R	Q with zero V i.e. Real
q	-2.6458 1.4142 -1.7321 2.0000	0 2 3 4	2 0 0 0
Inv(q)	-0.1654 -0.0884 0.1083 -0.1250	0 -0.0690 -0.1034 -0.1379	0.5000 0 0 0
q*inv(q)	1 0 0 0	1.0000 0 0 0	1.0000 0 0 0
inv(q)*q	1.0000 0 0.0000 0	1.0000 0 0 0	1.0000 0 0 0
	q*inv(q) is equal to inv(q)*q		
inv(inv(q))	-2.6458 1.4142 -1.7321 2.0	0 2 3 4	2 0 0 0
q - inv(inv(q))	0 0 0 0	0 0 0 0	0 0 0 0
	inv{inv(xQ)} is equal to Q ; since, (abs(inv_invxq-xq))< eps (=2.22e-16)		

5. Quaternion algebra

ॐ Addition

Adding quaternions is simpe; one just adds the corresponding multipliers (Formula. 6). Subtraction, in effect is addition by scalar multiplication of q2 with -1. ML_Fn 05 illustrates with examples operable inadvertently by looking at them.

ML_Fn 05: Addition of two quaternions

Formulae

Addition of two quaternions (q1, q2)

Input

$$q1 = a1 + b1 * i + c1 * j + d1 * k; \quad q1 = [a1, v1]$$

$$q2 = a2 + b2 * i + c2 * j + d2 * k; \quad q2 = [a2, v2]$$

Addition

$$q1 + q2 = a1 + b1 * i + c1 * j + d1 * k + a2 + b2 * i + c2 * j + d2 * k$$

$$= (a1 + a2) + i * (b1 + b2) + j * (c1 + c2) + k * (d1 + d2)$$

or

$$q1plusq2 = [(a1 + a2)(v1 + v2)] \quad Formula.6$$

```
%
%   om_addQuat.m (R S Rao) 6/11/16
%
 function [q1plusq2]= om_addQuat
(AugendQuat,AddendQuat)
%

if nargin ==0
  AugendQuat =  [0.1235,1  1 1 ]
  AddendQuat =  [0.8765,1,-1,0 ]
end
%
AddOp= 'q1plusq2 =  [AugendQuat + AddendQuat];'; %
Formula. 6
eval(AddOp)
disp(AddOp)
```

```
>> [sumQuat]= om_addQuat
```

Table 6: Sum of two quaternions

AugendQuat =	0.1235	1.0000	1.0000	1.0000
AddendQuat =	0.8765	1.0000	-1.0000	0
q1plusq2 =	1	2	0	1
q1plusq2 = [AugendQuat + AddendQuat];				

```
q1 = [9:-1:6]
q2 = [8:-1:5]
[q1minusq2]=om_addQuat (q1,-q2)
```

Table 6b: subtraction of two quaternions

q1 =	9 8 7 6	q1	9 8 7 6
q2 =	8 7 6 5	-q2	-8 -7 -6 -5
q1minusq2=	1 1 1 1	q1Plus minusq2	1 1 1 1

ॐ Multiplication

The product of two quaternions (q1 and q2) called the Hamilton product is calculated by the products of the basis elements and the distributive law (ML_Fn06, table 7a).

ML_Fn 06: Product of two quaternions (q1, q2)

$$q1 = a1 + b1*i + c1*j + d1*k$$
$$q2 = a2 + b2*i + c2*j + d2*k$$
$$q1*q2 = (a1 + b1*i + c1*j + d1*k)*$$
$$(a2 + b2*i + c2*j + d2*k$$
$$= p1 + i*p2 + j*p3 + k*p4$$

$$p1 = a1*a2 - b1*b2 - c1*c2 - d1*d2;$$
$$p2 = a1*b2 + b1*a2 + c1*d2 - d1*c2;$$
$$p3 = a1*c2 - b1*d2 + c1*a2 + d1*b2;$$
$$p4 = a1*d2 + b1*c2 - c1*b2 + d1*a2;$$

```
>> [prodQuat]= om_prodQuat
```

Table 7a: Product (q1*q2)

q1	1 1 1 1
q2	1 2 3 4
q1*q2	-8 4 2 6

```
%
%  om_prodQuat.m  (R S Rao 12-11-15)
%
function [prodQuat]=
om_prodQuat(xq1,xq2)
DefaultInpQuatProd
%
% Coeffients picking up
%
a1 = xq1(1,1);b1 = xq1(1,2);c1 =
xq1(1,3);d1 = xq1(1,4);

a2 = xq2(1,1);b2 = xq2(1,2);c2 =
xq2(1,3);d2 = xq2(1,4);
xq1,xq2

%% Formulae for  product of two
quaternions (xq1*xq2)
%
p1  = a1*a2 - b1*b2 - c1*c2 - d1*d2;
p2  = a1*b2 + b1*a2 + c1*d2 - d1*c2;
p3  = a1*c2 - b1*d2 + c1*a2 + d1*b2;
p4 =  a1*d2 + b1*c2 - c1*b2 + d1*a2;
prodQuat = [p1,p2,p3,p4];
%%
```

Product of two quaternions is non-commutative: The multiplication of quaternions is not commutative (Table 7b, ML_Fn 06b). At the time of invention of quarter ions, this property was a radical feature. But, now non-commutative algebraic operations are quite common in matrix, tensor and vector computations.

ML_Fn 06b: Properties of product of quaternions

```
%
% Prop_prodQ.m    (R S Rao) 10/11/15; 10-7-05
%
function  [q1q2,q2q1]= Prop_prodQ(xq1,xq2,type)
if nargin ==0
   DefaultInpQuatProd
end
if nargin <3
    type = ' ';
```

```
end
[q1q2]= om_prodQuat(xq1,xq2);
[q2q1]= om_prodQuat(xq2,xq1);
differences = abs ([q1q2 - q2q1]);

 %
 %  Numerical knowledge based inference
  blank   = setstr(ones(1,9)*' ');
 Ant{:,1} = 'differences < 1e-10' ;
 conseq{:,1} =  [blank,'!!!! Multiplication of ',type,' is not commutative !!!,
Since at least one of '];
 conseq{:,2} =  [ blank, ' $$$ Multiplication of ',type,' is commutative  $$$,
Since all'];
 Reason{:,1} = [blank,'(between q1*q2 and q2*q1)'];
 Reason{:,2} = [blank,'            > 1e-10'];
 Reason{:,3} = [blank,'            < 1e-10'];
   format short e
     if (eval(Ant{:,1}))
         disp(conseq{:,2}),differences, disp(Reason{:,1}),disp(Reason{:,3})
     else
         disp(conseq{:,1}),
         differences, disp(Reason{:,1}), disp(Reason{:,2})
     end
 format
```

Table 7b: Multiplication of two quaternions (Non commutative)

	Unequal quaternions ~~~~~~~~~~~~~~~~~~~~~~~~~	
q1	1 1 1 1	!!!! Multiplication of is not commutative!!! Since at least one of differences = [0 2 4 2] between q1*q2 and q2*q1 > 1e-10
q2	1 2 3 4	
q1*q2	-8 4 2 6	
q2*q1	-8 2 6 4	
Abs(diff)	0 2 4 2	

Auto test module for product operation: Here Prop_prodQ.m object module in matlab is tested with complex, scalar real and quaternions even with zero real components (table 7c) using AutoTest_prod.m (ML_Fn 06c). The inferences drawn from a few knowledge bits are shown in the right side of table.

Table 7c: Results of AutoTest_Prod.m for Multiplication of different number systems

	complex numbers (a + ib) ~~~~~~~~~~~~~~~~~~~~~~~~~	
q1	1 2 0 0	$$$ Multiplication of complex (a + i*b) numbers is commutative $$$, Since all differences =0 0 0 0 (between q1*q2 and q2*q1) < 1e-10
q2	2 4 0 0	
q1*q2	-6 8 0 0	
q2*q1	-6 8 0 0	
	Real number with no complex part ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~	
q1	4 0 0 0	Multiplication of Real number with no complex part is

q2 2 0 0 0 q1*q2 8 0 0 0 q2*q1 8 0 0 0	commutative Since all differences = [0 0 0 0] (between q1*q2 and q2*q1) < 1e-10
% Real numbers (zero) ~~~~~~~~~~~~~~~~~~~ q1 0 0 0 0 q2 0 0 0 0 q1*q2 0 0 0 0 q2*q1 0 0 0 0	Multiplication of Real numbers (zero) is commutative Since all differences = [0 0 0 0] (between q1*q2 and q2*q1) < 1e-10
Equal Quaternions ~~~~~~~~~~~~~~~~~~~~~~~~~~ q1 1 2 3 4 q2 1 2 3 4 q1*q2 -28 4 6 8 q2*q1 -28 4 6 8 Abs(diff) 0 0 0 0	Multiplication of Equal Quaternions is commutative Since all differences = [0 0 0 0] (between q1*q2 and q2*q1) < 1e-10
Quaternion and its inverse q1 4 3 2 1 q2 0.1333 -0.1000 -0.0667 -0.0333 q1*q2 1.0000 0 0 -0.0000 q2*q1 1.0000 0 0 -0.0000 Abs(diff) 0 0 2.7756e-17 2.7756e-17	Multiplication of Quaternion and its inverse is commutative Since all differences = [0 0 2.7756e-17 2.7756e-17] between q1*q2 and q2*q1 < 1e-10

ML_Fn 06c: Auto testing of multiplication of quaternions, imaginary and scalar real numbers

```
% AutoTest_prod.m
%
function AutoTest_prod(q1,q2)
if nargin == 2
    [q1q2,q2q1]= Prop_prodQ(q1,q2);
    return
end
[q1q2,q2q1]= Prop_prodQ
type = '   complex numbers (a + i*b)';
dispst(type)
xq1 = [1      2      0      0];
xq2 = [2      4      0      0];
[q1q2,q2q1]= Prop_prodQ(xq1,xq2,type)
%
type = ' Real numbers with no complex
part';
dispst(type)
xq1 = [4      0      0      0];
xq2 = [2      0      0      0];
[q1q2,q2q1]= Prop_prodQ(xq1,xq2,type)
%
type = '% Octonion';
dispst(type)
xq1 = [1:8];
xq2 = 2*[1:8];
[q1q2,q2q1]= Prop_prodQ(xq1,xq2,type)
```

```
%
%    exitom.m (R S Rao 18/12/05), 22/7//13
function exitom(mfileName)
if nargin < 1
    mfileName = 'Om'
end
whiteSpace= setstr(ones(1,18)*' ');

st = [whiteSpace,'exit from ',mfileName,
'.m','    ------------>'];
disp(' '),disp(st),disp(' ')
```

```
%
%    calledom.m (R S Rao 18/12/05)22/7//13
function calledom(mFileName)
if nargin < 1
    mFileName = 'Om'
end
%
whiteSpace18= setstr(ones(1,18)*' ');
%
st = [whiteSpace18,' =======>  Calling
',mFileName, '.m'];
disp(' '),disp(st),disp(' ')
```

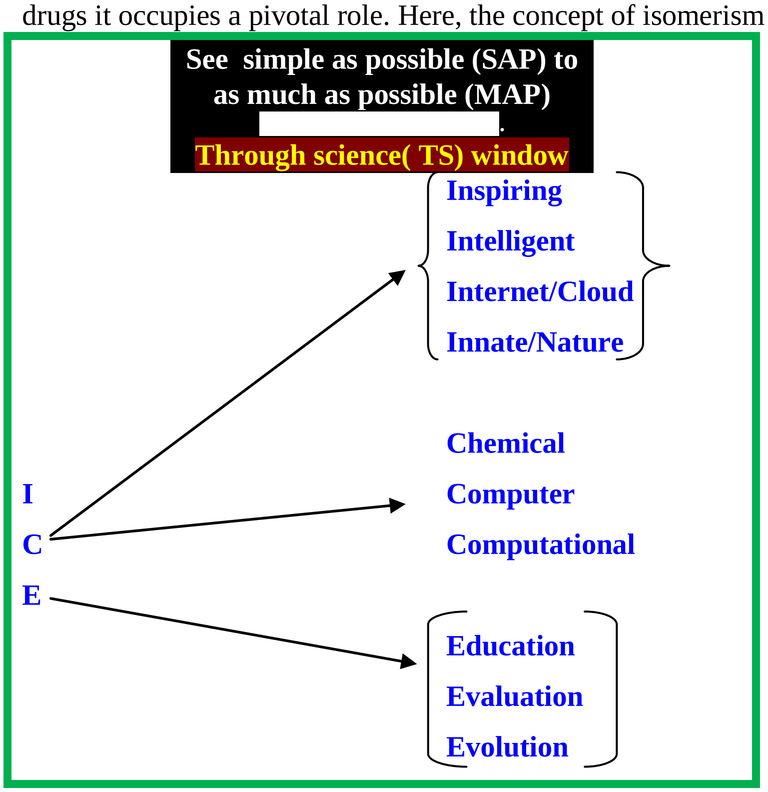
```
 type = '  Real numbers (zero)';
 dispst(type)
xq1 = [0      0      0      0];
xq2 = [0      0      0      0];
[q1q2,q2q1]= Prop_prodQ(xq1,xq2,type)
%
type = 'Equal Quaternions '; %
 dispst(type)
 xq1 = [1  2 3 4];
 xq2 = [1  2 3 4];
 [q1q2,q2q1]= Prop_prodQ(xq1,xq2,type)
%
 type = 'Equal Quaternion with zero real
part'; %
 dispst(type)
 xq1 = [0  2 3 4];
 xq2 = [0  3 4 5];
 [q1q2,q2q1]= Prop_prodQ(xq1,xq2,type)
 %
 type = 'Quaternion and its inverse '; %
 dispst(type)
 xq = [4 3 2 1];
 [invXq]= om_inverseQuat(xq);
 format short e
 [q1q2,q2q1]= Prop_prodQ(xq,invXq,type)
```

```
%
% dispst.m  (R S Rao 19/12/05)
%
function dispst(st)
if nargin ==0
    st = ' ';
end
center02(st,2,40);
```

6.0 Supplementary Material

ॐ **Inspiring chemical education (Ice) --Isomers**

In chemistry isomerism is a sought after and well-nourished discipline. Now, in biological molecules and drugs it occupies a pivotal role. Here, the concept of isomerism is mapped to virtual 'ICE'.



36 isomers

Brain integrates knowledge when one is at rest (in lighter moments)

It does not mean everything is seen (in nature)

```
%
%    ICE99.m   [RSRao 31-12-15] [15-5-13; 6-12-09]
%
 clean
  format compact
 I = {'inspiring'; 'Intelligent'; 'Internet'; 'Innate';};
 C = {'Chemical' ; 'Computer'; 'Computational';};
 E = {'Education';'Evaluation' ; 'Evolution' ; };
ws = ' '; k = 0;
 for j1 = 1:4
    for j2 = 1:3
       for j3 = 1:3
          ICE{j1,j2,j3} = [I{j1,:},ws,C{j2,:},ws, E{j3,:}];
          ICEzz = [I{j1,:},ws,C{j2,:},ws, E{j3,:}];
          disp(ICEzz)
          k = k+1;
       end
    end
 end
```

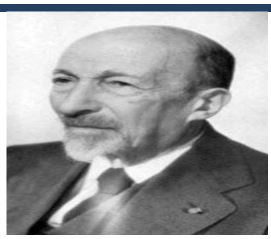
Isomers of $_{\text{Acronym}}$ 'ICE'
~~~~~~~~~~~~~~~~~~~~~~~~~~~~

1 inspiring Chemical Education
2 inspiring Chemical Evaluation
3 inspiring Chemical Evolution
4 inspiring Computer Education
5 inspiring Computer Evaluation
6 inspiring Computer Evolution
7 inspiring Computational Education
8 inspiring Computational Evaluation
9 inspiring Computational Evolution

10 Intelligent Chemical Education
11 Intelligent Chemical Evaluation
12 Intelligent Chemical Evolution
13 Intelligent Computer Education
14 Intelligent Computer Evaluation
15 Intelligent Computer Evolution
16 Intelligent Computational Education
17 Intelligent Computational Evaluation
18 Intelligent Computational Evolution

19 Internet/Cloud Chemical Education
20 Internet/Cloud Chemical Evaluation
21 Internet/Cloud Chemical Evolution
22 Internet/Cloud Computer Education
23 Internet/Cloud Computer Evaluation
24 Internet/Cloud Computer Evolution
25 Internet/Cloud Computational Education
26 Internet/Cloud Computational Evaluation
27 Internet/Cloud Computational Evolution

28 Innate Chemical Education
29 Innate Chemical Evaluation
30 Innate Chemical Evolution
31 Innate Computer Education
32 Innate Computer Evaluation
33 Innate Computer Evolution
34 Innate Computational Education
35 Innate Computational Evaluation
36 Innate Computational Evolution



Jacques Hadamard
1865–1963
French mathematician

The shortest path between two truths in the real domain passes through the complex domain.

REFERENCES

Applications-Quaternions

[1] T. Jiang, X. Cheng, S. Ling, *Applied Mathematics Letters*, **2016,** 52, *58-63.*
An algebraic technique for total least squares problem in quaternionic quantum theory
~~~~~~~~~~~~~~~~~~~~~~~~~~~~

[2] *J. A. D. García, F. J. C. Lopera, J. Multivariate Analysis,* **2016,** 144, *139-149.*
Elliptical affine shape distributions for real normed division algebras
[3] M. Melander, K. Laasonen and H. Jónsson, *J. Chem. Theory Comput.*, **2015,** 11, 3, *1055–1062.*
Removing External Degrees of Freedom from Transition-State Search Methods using Quaternions
[4] Z. Zhang, Z. Jiang, T. Jiang, *Appl. Math. Comput.*, **2015,** 269, 15, *618-625.*
Algebraic methods for least squares problem in split quaternionicmechanics
[5] A. Rehman, Q. Wen W. Z. H. He, *Appl. Math. Comput.*, **2015,** 265, 15, *945-957.*
Solution to a system of real quaternion matrix equations encompassing η-Hermicity
[6] S.T. Ling, X. H. Cheng, T. S. Jiang, *Appl. Math. Comput.*, **2015,** 270, 1, *984-992.*
Consimilarity of quaternions and coneigenvalues of quaternion
[7] F. Zhang, M. Wei, Y. Li, J. Zhao, *Appl. Math. Comput.*, **2015,** 270, 1, *425-433.*
Special least squares solutions of the quaternion matrix equation AX=B with applications
[8] S. J. Sangwine, *Comput. Phys. Commun.*, **2015,** 188, *128-130.*
Comments on 'A structure-preserving method for the quaternion LU decomposition in quaternionic quantum theory' by Minghui Wang and Wenhao Ma
[9] J. A. D. García, F. J. C. Lopera, *J. King Saud University - Science,* **2015**.
Matric variate Pearson type II-Riesz distribution
[10] K. Majewski, D. Ritter, *J. Mag. Resonance,* **2015,** 258, *65-80.*
First and second order derivatives for optimizing parallel RF excitation waveforms
[11] L.P. Chen, K. I. Kou, M. S. Liu, *J. Math. Analysis and Applications*, **2015,** 423, 1, *681-700.*
Pitt's inequality and the uncertainty principle associated with the quaternion Fourier transform
[12] D. Schulz, R. S. Thomä, *Signal Processing,* **2015,** 108, *245-258.*
Quaternion-based polarimetric array manifold interpolation
[13] Y. Li, M. Wei, F, Zhang, J. Zhao, *Appl. Math. Comput.*, **2014,** 25, *157-167.*
A fast structure-preserving method for computing the singular value decomposition of quaternion matrices
[14] T. Jiang, Z. Jiang, S. Ling, *Appl. Math. Comput.*, **2014,** 249, 15, *222-228.*
An algebraic method for quaternion and complex Least Squares coneigen-problem in quantum mechanics
[15] J. Prošková, *J. Molecular Structure*, **2014,** 1076, *89-93.*
Description of protein secondary structure using dual quaternions
[16] M. Wang, W. Ma, *Appl. Math. Comput.*, **2013,** 223, 15, *354-361.*
A structure-preserving algorithm for the quaternion Cholesky decomposition
[17] M. Wang, W. Ma, *Comput. Phys. Commun.*, **2013,** 184, 9, *2182-2186.*
A structure-preserving method for the quaternion LU decomposition in quaternionic quantum theory
[18] G. Skone, S. Cameron, and I. Voiculescu, *J. Chem. Inf. Model.*, **2013,** 53, 12, *3367–3372.*
Doing a Good Turn: The Use of Quaternions for Rotation in Molecular Docking
[19] Z. Jia, M. Wei, S. Ling, *J. Comput. Appl. Math.*, **2013,** 239, 1, *12-24.*
A new structure-preserving method for quaternion Hermitian eigenvalue problems
[20] I. Kyrchei, *Linear Algebra and its Applications,* **2013,** 438, 1, *136-152.*
Explicit representation formulas for the minimum norm least squares solutions of some quaternion matrix equations
[21] J. S. Anderson, D. M. LeMaster, *Biophysical Chemistry,* **2012,** *168–169, 28-39.*
Rotational velocity rescaling of molecular dynamics trajectories for direct prediction of protein NMR relaxation
[22] T. Hynninen, C.L. Dias, A. Mkrtchyan, V. Heinonen, M. Karttunen, A.S. Foster, T. A. Nissila, *Comput. Phys. Commun.*, **2012,** 183, 2, *363-369.*
A molecular dynamics implementation of the 3D Mercedes-Benz water model
[23] S. Caiqin, C. Guoliang, W. Xiaodong, *cta Mathematica Scientia, Volume,* **2012,** 32, 5, *1967-1982.*
On solutions of quaternion matrix equations XF # AX = BY and XF # A = BY

[24] D, Sehnal, R. S. Vařeková, H. J. Huber, S. Geidl, C. M. Ionescu, M. Wimmerová, and J. Koča, *J. Chem. Inf. Model.*, **2012,** 52, 2, *343–359.*
SiteBinder: An Improved Approach for Comparing Multiple Protein Structural Motifs

[25] J. Romanowska, K. S. Nowiński, and J. Trylska, *J. Chem. Theory Comput.*, **2012,** 8, 8, *2588–2599.*
Determining Geometrically Stable Domains in Molecular Conformation Sets

[26] A. J. Hanson, S. Thakur, *J. Molecular Graphics and Modelling*, **2012,** 38, *256-278.*
Quaternion maps of global protein structure

[27] M. Bahri, R. Ashino, R.Vaillancourt, *Appl. Math. Comput.*, **2011,** 217, 1, *10-21.*
Two-dimensional quaternion wavelet transform

[28] D. T. Murray, Y. Lu, T.A. Cross, J.R. Quine, *J. Mag. Resonance*, **2011,** 210, 1, *82-89.*
Geometry of kinked protein helices from NMR data

[29] I. A. Khan, Q.W. Wang, G. J. Song, *Appl. Math. Comput.*, **2010,** 217, 5, *2031-2040.*
Minimal ranks of some quaternion matrix expressions with applications

[30] M. Wang, *Comput. Phys. Commun.*, **2010,** 181, 6, *1047-1050.*
Algorithm Q-LSQR for the least squares problem in quaternionic quantum theory

[31] S. Ling, M. Wang, M. Wei, *Comput. Phys. Commun.*, **2010,** 181, 3, *481-488.*
Hermitian tridiagonal solution with the least norm to quaternionic least squares problem

[32] E. Abrahamsson, S. S. Plotkin, *J. Molecular Graphics and Modelling*, **2009,** 28, 2, *140-145.*
BioVEC: A program for Biomolecule Visualization with Ellipsoidal Coarse-graining

[33] L. Borsten, D. Dahanayake, M.J. Duff, H. Ebrahim, *W. Rubens*, *Phys. Reports*, **2009,** 471, 3-4, *113-219.*
Black holes, qubits and octonions

[34] T. Jiang, L. Chenss, *Comput. Phys. Comm.*, **2008,** 178, 11, 795-799.
An algebraic method for Schrödinger equations in quaternionic quantum mechanics

[35] M. Wang, M. Wei, Y. Feng, *Comput. Phys. Commun.*, **2008,** 179, 4, *203-207.*
An iterative algorithm for least squares problem in quaternionic quantum theory

[36] T. Jiang, L. Chen, *Comput. Phys. Commun.*, **2008,** 178, 11, *795-799.*
An algebraic method for Schrödinger equations in quaternionic quantum mechanics

[37] A. V. Akimov, A. V. Nemukhin, A. A. Moskovsky, A. B. Kolomeisky and J. M. Tour, *J. Chem. Theory Comput.*, **2008,** 4, 4, *652–656.*
Molecular Dynamics of Surface-Moving Thermally Driven Nanocars

[38] P. B. Slater, *J. Geometry and Physics*, **2008,** 58, 9, *1101-1123.*
Extended studies of separability functions and probabilities and the relevance of Dyson indices

[39] *T. Jiang, L. Chen, Comput. Phys. Commun.*, **2007,** 176, 7, *481-485.*
Algebraic algorithms for least squares problem in quaternionic quantum theory

[40] C. F.F. Karney, *J. Molecular Graphics and Modelling*, **2007,** 25, 5, *595-604.*
Quaternions in molecular modeling

[41] K. Kazimierczuk, W. Koźmiński, I. Zhukov, J. Mag. Resonance, **2006,** 179, 2, 323-328.
Two-dimensional Fourier transform of arbitrarily sampled NMR data sets

[42] T. Saue, *Advances in Quantum Chemistry*, **2005,** 48, *383-405.*
Spin-Interactions and the Non-relativistic Limit of Electrodynamics

[43] Y. Sun,W. J. Welsh and R. A. Latour, *Langmuir*, **2005,** 21, 12, *5616–5626.*
Prediction of the Orientations of Adsorbed Protein Using an Empirical Energy Function with Implicit Solvation

[44] *S. D. Leo, G.C. Ducati, Comput. & Mathematics with Applications*, **2004,** 48, 12, *1893-1903.*
Real linear quaternionic differential operators

[45] H. P. Fritzer, *Spectrochimica Acta Part A: Molecular and Biomolecular Spectroscopy*, **2001,** 57, 10, *1919-1930.*
Molecular symmetry with quaternions

[46] *J.R. Quine, J. Molecular Structure: THEOCHEM*, **1999,** 460, 1-3, *53-66.*
Helix parameters and protein structure using quaternions

[47] R. Abłamowicz, *Comput. Phys. Commun.*, **1998,** 115, 203, *510-535.*
Spinor representations of Clifford algebras: a symbolic approach
[48] S.H. Walmsley, *J. Molecular Structure*, **1998,** 189, 1-2, *129-135.*
A comparison of infinitesimal rotation, Quaternion and direction cosine coordinates in the dynamics of molecular clusters
[49] D. P. Stapleton, *Comput. Phys. Commun.*, **1996,** *98*, 1-2, *153-166.*
Numerical simulation of a rigid rotating body by Obrechkoff integration
[50] D. Lu, S. Ali, D. J. Siminovitch, *J. Mag. Resonance, Series A*, **1996,** 122, 2, *192-203.*
A Quaternion Analysis of Time Symmetry in Spin-1 Excitation
[51] J. Shen, J. Mag. Resonance, Series A, **1995,** 117, 1, 98-102.
A Simple Quaternion Analysis of Selective Decoupling Pulses
[52] T. F. Havel, I. Najfeld, *J. Molecular Structure: THEOCHEM*, **1995,** 336, 2-3, *175-189.*
Applications of geometric algebra to the theory of molecular conformation 2. The local deformation problem
[53] L. Emsley, G. Bodenhausen, *J. Mag. Resonance*, **1992,** 97, 1, *135-148.*
Optimization of shaped selective pulses for NMR using a quaternion description of their overall propagators
[54] B Blümich, H.W Spiess, J. Mag. Resonance, **1985,** 61, 2, 356-362.
Quaternions as a practical tool for the evaluation of composite rotations
[55] B.R. Markey, S.H. Walmsley, *Chemical Phys. Letters*, **1982,** 92, 3, *257-261.*
Quaternion formulation of lattice dynamics of molecular crystals
[56] L.L. Boyle, *Chemical Phys. Letters*, **1972,** 13, 3, *331-333.*
Quaternionic wavefunctions for triply degenerate states

Quaternions-Hamilton

[57] W. R. Hamilton, Philosophical Magazine., **1844**, 25(3),489–495.
On quaternions or on a new system of imaginaries in algebra.
[58] William Rowan Hamilton, **1853.**
"*Lectures on Quaternions*", Royal Irish Academy.
[59] W.R. Hamilton, **1866.**
Elements of Quaternions University of Dublin Press. Edited by William Edwin Hamilton, son of the deceased author.
[60] W.R. Hamilton, **1899.**
Elements of Quaternions volume I, (1901) volume II. Edited by Charles Jasper Joly; published byLongmans, Green & Co.

AUTHORS' ADDRESSES

1. **K RamaKrishna**
Department of Chemistry,
Gitam Institute of Science,
Gitam University,
Visakhapatnam, A.P
E-mail: karipeddirk@gmail.com
2. **R. Sambasiva Rao**
School of Chemistry,
Andhra University,
Visakhapatnam 530 003, A.P
E-mail: rsr.chem@gmail.com