RepositóriUM
Universidade do Minho

CMAT
Centro de Matemática da Universidade do Minho
Campus de Gualtar 4710-057 Braga Portugal
www.cmat.uminho.pt

Universidade do Minho
Escola de Ciências
Centro de Matemática

Quaternions: a Mathematica Package for Quaternionic Analysis

M.I. Falcão[a] Fernando Miranda[b]

[a]Departamento de Matemática e Aplicações, Universidade do Minho, Portugal

[b]Departamento de Matemática e Aplicações and Centro de Matemática, Universidade do Minho, Portugal

Information

Keywords:
Quaternions, Clifford Analysis, monogenic functions, symbolic computation.

Original publication:
Computational Science and its Applications, Lecture Notes in Computer Science, vol. 6784, pp. 200-214, 2011
DOI: 10.1007/978-3-642-21931-3_17
www.springerlink.com

Abstract

This paper describes new issues of the Mathematica standard package `Quaternions` for implementing Hamilton's Quaternion Algebra. This work attempts to endow the original package with the ability to perform operations on symbolic expressions involving quaternion-valued functions. A collection of new functions is introduced in order to provide basic mathematical tools necessary for dealing with regular functions in $\mathbb{R}^{n+1}$, for $n \geq 2$. The performance of the package is illustrated by presenting several examples and applications.

1 Introduction

Quaternions were introduced in 1843 by the Irish mathematician William Rowan Hamilton. One of the most popular application of Hamilton's Algebra is concerned with the use of quaternions for describing 3D rotations. In fact, *quaternions are inextricably linked to rotations* ([4]) and their use has become indispensable in all high technologies with need of calculations in real time.

Nowadays, with the development of Quaternionic Analysis, quaternions are also recognized as a powerful tool for modeling and solving problems in both theoretical and applied mathematics ([20]).

The increasing interest in using quaternions and their applications in almost all applied sciences has motivated the emergence of several software packages to perform computations in the algebra of the real quaternions (see, for example, [15,16,22]), or more generally, in Clifford Algebras (see [1,3] and the references therein for details).

Three main reasons lead us to develop this work:

- to endow the standard package `Quaternions` with the ability to perform operations on quaternion-valued functions;
- to extend the applicability of the package to arbitrary dimensions;
- to introduce a basic set of special polynomials, which plays an important role in applications.

Mathematica is a registered trademark of Wolfram Research, Inc.

2 Algebra of Quaternions

2.1 Basic Results

Any quaternion x can be written in the form

$$x = x_0 + ix_1 + jx_2 + kx_3,\ x_i \in \mathbb{R}\,, \tag{1}$$

where Hamilton's imaginary units i, j and k satisfy the multiplication rules

$$i^2 = j^2 = k^2 = -1 \text{ and } ij = -ji = k\,. \tag{2}$$

This non-commutative product generates the algebra of real quaternions $\mathbb{H}$. The real vector space $\mathbb{R}^4$ will be embedded in $\mathbb{H}$ by identifying the element $x = (x_0, x_1, x_2, x_3) \in \mathbb{R}^4$ with the element $x = x_0 + ix_1 + jx_2 + kx_3 \in \mathbb{H}$. Thus, throughout this paper, we will use the same symbol x to represent a point in $\mathbb{R}^4$ and the corresponding quaternion in $\mathbb{H}$.

For a quaternion x of the form (1) we will distinguish between the real part of x,

$$\operatorname{Re} x := x_0\,,$$

and the vector part of x,

$$\operatorname{Vec} x = \underline{x} := ix_1 + jx_2 + kx_3\,,$$

so that a quaternion x can be written as

$$x = x_0 + \underline{x}\,.$$

When $x = \underline{x}$, x is called a *pure quaternion*. The conjugate of x is

$$\bar{x} := x_0 - \underline{x}$$

and the norm of x, $|x|$, is defined by

$$|x|^2 = x\bar{x} = \bar{x}x = x_0^2 + x_1^2 + x_2^2 + x_3^2\,.$$

If $|x| = 1$, x is said to be a *unit quaternion*. It immediately follows that each non-zero $x \in \mathbb{H}$ has an inverse given by

$$x^{-1} = \frac{\bar{x}}{|x|^2}$$

and therefore $\mathbb{H}$ is a non-commutative division ring or a skew field.

We note that an arbitrary non-null quaternion x can be written as

$$x = x_0 + \omega(x)|\underline{x}|\,, \tag{3}$$

where $\omega(x)$ is the unit quaternion

$$\omega(x) = \frac{\underline{x}}{|\underline{x}|}\,, \tag{4}$$

very much like a complex number is written in the form $a + ib$.[1] Moreover, since $\omega^2 = -1$, ω behaves like the imaginary unit. In fact, if $x = x_0 + \omega(x)|\underline{x}|$ and $y = y_0 + \omega(y)|\underline{y}|$ are quaternions such that $\omega(x) = \omega(y) = \omega$, then all the algebraic operations can be computed as if x and y were complex numbers, in particular

$$xy = yx = (x_0 + \omega|\underline{x}|)(y_0 + \omega|\underline{y}|) = x_0y_0 - |\underline{x}||\underline{y}| + \omega(x_0|\underline{y}| + |\underline{x}|y_0)\,.$$

[1] In literature concerning quaternionic treatment of rotations, (3) and (4) are commonly referred to as the *binary form* of x and the *sign* of $\underline{x}$, respectively.

2.2 Additional Functions

`Quaternions` is a Mathematica standard package to implementing Hamilton's Quaternion Algebra. It adds rules to `Plus`, `Minus`, `Times`, `Divide` and the fundamental `NonCommutativeMultiply`. Among others, the following quaternion functions are included: `Re`, `Conjugate`, `Abs`, `AbsIJK`, `Norm`, `Sign`, `AdjustedSignIJK`, `ToQuaternion`, `FromQuaternion` and `QuaternionQ`. Help on the use of these functions can be obtained from the Quaternions package guide. In the package, a quaternion is an object of the form `Quaternion[x0,x1,x2,x3]`. In the original version of the package, quaternions must have real numeric valued entries. This extended version allows the use of symbolic entries, assuming that all symbols represent real numbers.

The complex-like representation (3) of a quaternion is quite useful and thus we introduce an object of the form `ComplexLike[a,b]`. For such objects, simple rules as `Plus`, `Times`, `Power` and functions as `Re`, `Abs`, `Norm`, etc. are defined.

The main commands to perform algebraic operations on quaternions are essentially the original ones. There are just some new commands:

`Vec`	gives the vector part of a quaternion.
`PureQuaternionQ`	gives True for pure quaternions and False otherwise.
`W`	gives the sign of a quaternion.
`ToComplexLike`	returns a quaternion in the complex-like form.
`QPower`	recursive implementation of the Power.

Example 1. Simple Functions

`In[1]:= q1=Quaternion[1,2,-2,-1];`

`In[2]:= Vec[q1]`

`Out[2]= Quaternion`$[0,\ 2,\ -2,\ -1]$

`In[3]:= % // PureQuaternionQ`

`Out[3]= True`

Example 2. The `ComplexLike` object

`In[4]:= q2=Quaternion[x0,x1,x2,x3];`

`In[5]:= ToComplexLike[q2]`

`Out[5]= ComplexLike` $\left[\mathtt{x0}, \sqrt{\mathtt{x1}^2+\mathtt{x2}^2+\mathtt{x3}^2}\right]$

`In[6]:= % // TraditionalForm`

`Out[6]=` $\mathtt{x0}+\sqrt{\mathtt{x1}^2+\mathtt{x2}^2+\mathtt{x3}^2}\omega$

Example 3. Operations on `ComplexLike` objects

`In[7]:= q3 = Quaternion[-1, 4, -4, -2];`

`In[8]:= clq1 = ToComplexLike[q1]`

`Out[8]= ComplexLike[1, 3]`

`In[9]:= clq3 = ToComplexLike[q3]`

`Out[9]= ComplexLike[-1, 6]`

`In[10]:= {q1**q3 // ToComplexLike,clq1*clq3}`

`Out[10]= {ComplexLike[-19, 3],ComplexLike[-19, 3]}`

`In[11]:= {Abs[clq1], Abs[q1]}`

`Out[11]=` $\{\sqrt{10}, \sqrt{10}\}$

The rules for `Power` contained in the original package are based essentially on Moivre's theorem for quaternions and are more efficient when the quaternion is numeric valued. Here we adopt the power recursive implementation given by `QPower`.

Example 4. The `QPower` function

```
In[12]:= QPower[q1,3]
Out[12]= Quaternion[-26, -12, 12, 6]
In[13]:= QPower[q2,2]
Out[13]= Quaternion[x0^2 - x1^2 - x2^2 - x3^2, 2x0x1, 2x0x2, 2x0x3]
```

3 Quaternionic Analysis

3.1 The Concept of $\mathbb{H}$-regular Functions

In complex function theory there are three distinct, but equivalent, approaches to regular functions: Cauchy's approach connected with the notion of complex differentiability, the Weierstrass approach based on the use of convergent power series and the so-called Cauchy-Riemann equations, introduced by Riemann. Since $\mathbb{H}$ is a skew field, it is natural to ask whether differentiability of a function $f : \mathbb{H} \to \mathbb{H}$ can be defined in a similar way as in the cases of $\mathbb{R}$ and $\mathbb{C}$. The functions of a quaternion variable which have quaternionic derivatives, in the natural sense, are just the constant and linear functions (and not all of them); the functions which can be represented by quaternionic power series are those which can be represented by power series in four real variables. The first approach leads to a very restrictive class of *regular* functions, while the second one gives a too large class of functions. See [15] and the references therein for details and also [23].

In 1935, R. Fueter, one of the founders of Quaternionic Analysis ([11,12]), proposed a generalization of complex analyticity to the quaternionic case by means of an analogue of the Cauchy-Riemann equations. He showed that this definition leads to close analogues of several important results from classical complex function theory ([23]). We describe briefly Fueter's approach to what he called *regular* $\mathbb{H}$-valued functions.

Consider a quaternion-valued function f of one quaternion variable x, defined in a domain $\Omega \subset \mathbb{R}^4$

$$f : \Omega \to \mathbb{H}\,,$$

$$f(x) = f_0(x) + i f_1(x) + j f_2(x) + k f_3(x)\,,$$

where $x = (x_0, x_1, x_2, x_3) \in \mathbb{R}^4$ and f_k are real valued in Ω functions. Continuity, differentiability or integrability are defined coordinate-wisely.

On the set $\mathscr{C}^1(\Omega, \mathbb{H})$ define the quaternionic Cauchy-Riemann operator

$$\overline{\partial} := \partial_0 + \partial_{\underline{x}}\,, \tag{5}$$

where $\partial_0 := \frac{\partial}{\partial x_0}$ and $\partial_{\underline{x}}$ is the Dirac operator

$$\partial_{\underline{x}} := i\frac{\partial}{\partial x_1} + j\frac{\partial}{\partial x_2} + k\frac{\partial}{\partial x_3}\,. \tag{6}$$

This leads to the following definition of $\mathbb{H}$-regular function or monogenic function (as called nowadays):

Definition 1 (Monogenic function). *A $\mathscr{C}^1$-function f satisfying the equation $\overline{\partial} f = 0$ (resp. $f\overline{\partial} = 0$) is called left monogenic (resp. right monogenic). A function which is both left and right monogenic is called monogenic.*

The concept of quaternionic or hypercomplex differentiability was first introduced by Malonek in [18,19]. Later on, the definition of hypercomplex derivative was generalized to higher dimensions [13].

Definition 2 (Hypercomplex derivative). *Let $f \in \mathscr{C}^1(\Omega, \mathbb{H})$ be a monogenic function in Ω. The hypercomplex derivative f' can be expressed by the real partial derivatives as*

$$f' = \frac{1}{2}\partial f\,, \tag{7}$$

where $\partial = \partial_0 - \partial_{\underline{x}}$ is the conjugate Cauchy-Riemann operator.

Since a hypercomplex differentiable function belongs to the kernel of $\overline{\partial}$, it follows that in fact

$$f' = \partial_0 f$$

like in the complex case. Obviously, last formula guarantees that the hypercomplex derivative of a monogenic function is again a monogenic function.

3.2 From Quaternionic Analysis to Clifford Analysis

Clifford Algebras were introduced in 1878 by the English geometer W. K. Clifford, generalizing the complex numbers and Hamilton's quaternions [8]. They have many applications to differential geometry, physics, robotics, computer vision, etc. (see, for example, [2]).

The foundation of Quaternionic Analysis by R. Fueter and his collaborators can be considered as the starting point of Hypercomplex Function Theory (as called by Fueter), or Clifford Analysis (as called nowadays).

Here we present the extension of the main definitions and results of the previous sections. Details about this subject and related topics can be found in [6,14,15].

Let $\{e_1, e_2, \cdots, e_n\}$ be an orthonormal basis of the euclidean vector space $\mathbb{R}^n$ with a product according to the multiplication rules

$$e_k e_l + e_l e_k = -2\delta_{kl},\ k,\, l = 1, \cdots, n\,,$$

where δ_{kl} is the Kronecker symbol. This non-commutative product generates the 2^n-dimensional Clifford Algebra $Cl_{0,n}$ over $\mathbb{R}$ and the set $\{e_A : A \subseteq \{1, \cdots, n\}\}$ with $e_A = e_{h_1}e_{h_2}\cdots e_{h_r}$, $1 \le h_1 < \cdots < h_r \le n$, $e_\emptyset = e_0 = 1$, forms a basis of $Cl_{0,n}$. Denoting by $\mathcal{A}_n$ the subset of the Algebra $Cl_{0,n}$,

$$\mathcal{A}_n := \text{span}_{\mathbb{R}}\{1, e_1, \ldots, e_n\}\,,$$

the real vector space $\mathbb{R}^{n+1}$ can be embedded in $\mathcal{A}_n$ by the identification of each element $(x_0, x_1, \cdots, x_n) \in \mathbb{R}^{n+1}$ with the *paravector* $x = x_0 + x_1e_1 + \ldots x_ne_n \in \mathcal{A}_n$.

Similarly to the quaternionic and complex case, a paravector can be written in terms of a real part and a vector part as $x = x_0 + \underline{x}$, the conjugate of x is $\bar{x} = x_0 - \underline{x}$ and the norm $|x|$ of x is defined by $|x|^2 = x\bar{x} = \bar{x}x = x_0^2 + x_1^2 + \cdots + x_n^2$. Moreover, denoting by $\omega(x) = \frac{\underline{x}}{|\underline{x}|} \in S^n$, where S^n is the unit sphere in $\mathbb{R}^n$, each paravector x can be written as a complex-like number

$$x = x_0 + \omega(x)|\underline{x}|\,. \tag{8}$$

In general, due to the algebraic properties of $Cl_{0,n}$, we have to assume that a monogenic function f, defined in some open subset $\Omega \subset \mathbb{R}^{n+1}$, has values in $Cl_{0,n}$, i.e., it is of the form $f(x) = \sum_A f_A(x)e_A$, where f_A are real functions.

The Cauchy-Riemann operator in $\mathbb{R}^{n+1}$ is obtained from the generalized Dirac operator

$$\overline{\partial} := \partial_0 + \partial_{\underline{x}}\,, \quad \text{where} \quad \partial_{\underline{x}} := \sum_{i=1}^{n} e_i \frac{\partial}{\partial x_i}$$

and f is a *monogenic function* in the sense of Clifford Analysis if it belongs to the kernel of $\overline{\partial}$. We suppose, once more, that f is hypercomplex differentiable in Ω in the sense of [18] and [13], i.e., $f' = \frac{1}{2}(\partial_0 - \partial_{\underline{x}})f = \partial_0 f$ like in the complex and quaternionic case.

3.3 New Functionalities

One of the objectives of this work is to endow the package with the ability to operate on paravector elements. For this purpose, the new object `Paravector` is introduced and some elementary operations are extended.[2]

If nothing is stated otherwise, it is assumed that $n = 3$ and therefore the new functions accept as arguments objects of the form `Quaternion` or `Paravector`.[3] For different values of n we have to declare the space dimension, through the command `CoordSys`.

Example 5. The `Paravector` object

```
In[14]:= CoordSys[5]
Out[14]= The coordinates system list is set to {X0,X1,X2,X3,X4}
In[15]:= Paravector[1,2,3,4,5] // TraditionalForm
```

Out[15]= $\mathrm{e}_0 + 2\,\mathrm{e}_1 + 3\,\mathrm{e}_2 + 4\,\mathrm{e}_3 + 5\,\mathrm{e}_4$

Example 6. Operations on `Paravector` objects

```
In[16]:= Paravector[1,2,3,4,5]+ 2 Paravector[5,4,3,2,1]
Out[16]= Paravector[11,10,9,8,7]
In[17]:= Conjugate[%]
Out[17]= Paravector[11,-10,-9,-8,-7]
In[18]:= W[Paravector[1,0,1,0,1]]
```

Out[18]= Paravector $\left[0, 0, \frac{1}{\sqrt{2}}, 0, \frac{1}{\sqrt{2}}\right]$

For `Quaternion` objects the package includes the following new functions:

CauchyRiemannL	left Cauchy-Riemann operator.
CauchyRiemannR	right Cauchy-Riemann operator.
DiracL	left Dirac operator.
DiracR	right Dirac operator.
MonogenicQ	gives True for monogenic functions.
Derivative	gives the derivative of monogenic functions.

We underline that the last two functions have been extended for `Paravector` and `ComplexLike` objects.

Example 7. Quaternion-valued functions

```
In[19]:= CoordSys[x0,x1,x2,x3]
Out[19]= The coordinates system list is set to {x0,x1,x2,x3}
In[20]:= f1=QPower[Quaternion[x0,x1,x2,x3],2]
```

Out[20]= Quaternion $\left[\mathrm{x0}^2 - \mathrm{x1}^2 - \mathrm{x2}^2 - \mathrm{x3}^2, 2\mathrm{x0x1}, 2\mathrm{x0x2}, 2\mathrm{x0x3}\right]$

```
In[21]:= CauchyRiemannR[f1]
Out[21]= Quaternion[-4x0,0,0,0]
```

In[22]:= f2=Quaternion $\left[\mathrm{x0}^2 - \frac{\mathrm{x1}^2}{3} - \frac{\mathrm{x2}^2}{3} - \frac{\mathrm{x3}^2}{3}, \frac{2\mathrm{x0x1}}{3}, \frac{2\mathrm{x0x2}}{3}, \frac{2\mathrm{x0x3}}{3}\right]$;

```
In[23]:= MonogenicQ[f2]
```

[2] The product of two paravectors in $\mathcal{A}_n$ is an element of $Cl_{0,n}$ but, in general, it is not an $\mathcal{A}_n$-element. Hence, the multiplication is not extended for this class of objects.

[3] We underline that, due to the algebraic properties of $\mathbb{H}$ and $\mathcal{A}_3$, a quaternion and a paravector in $\mathbb{R}^4$ have different natures.

```
Out[23]= True

In[24]:= Derivative[f2]
Out[24]= Quaternion[2x0, 2x1/3, 2x2/3, 2x3/3]

In[25]:= Derivative[f1]
         Quaternion::nonmonogenic: f is a non-monogenic function ≫
Out[25]= Derivative[Quaternion[x0^2 - x1^2 - x2^2 - x3^2, 2x0x1, 2x0x2, 2x0x3]]
```

Example 8. Paravector-valued functions

```
In[26]:= CoordSys[x,y,z]
Out[26]= The coordinates system list is set to {x,y,z}

In[27]:= p1=Paravector[x^2-y^2/2-z^2/2,x y,x z];
In[28]:= MonogenicQ[p1]
Out[28]= True

In[29]:= p2=Derivative[p1]
Out[29]= Paravector[2x, y, z]
```

Example 9. Complex-like functions

```
In[30]:= q1=Assuming[x r>0,Simplify[ToComplexLike[p1]/.y^2 → -z^2+r^2]]
Out[30]= ComplexLike[-r^2/2 + x^2, r x]

In[31]:= MonogenicQ[q1,x,r]
Out[31]= True

In[32]:= q2=Assuming[r>0,Simplify[ToComplexLike[p2]/.y^2 → -z^2+r^2]]
Out[32]= ComplexLike[2 x,r]

In[33]:= Derivative[q1,x,r]==q2
Out[33]= True
```

4 Generating Monogenic Functions

4.1 A Basic Set of Polynomials

In recent years, special hypercomplex Appell polynomials[4] have received attention from several authors and for different reasons ([5,7,17]).

In this section, we consider a basic set of polynomials first introduced in [9] for the 3-dimensional case and later on extended to higher dimensions in [10,21]. These polynomials can be written in terms of a paravector variable x and its conjugate as

$$\mathcal{P}_k^n(x) = \sum_{s=0}^{k} T_s^k(n)\, x^{k-s}\, \bar{x}^s, \;\; x \in \mathbb{R}^{n+1}, \;\; n \geq 1\,, \tag{9}$$

where

$$T_s^k(n) = \frac{k!}{n_{(k)}} \frac{\left(\frac{n+1}{2}\right)_{(k-s)} \left(\frac{n-1}{2}\right)_{(s)}}{(k-s)!s!}\,, \tag{10}$$

[4] We recall that a sequence of polynomials $P_0,\ P_1,\ldots$ is said to form an Appell sequence if: (i) P_k is of exact degree k, for each $k = 0,1,\ldots$; (ii)$P_k' = kP_{k-1}$, for each $k = 1,2,\ldots$. Examples of Appell sequences, besides the monomial functions, are the Hermite polynomials, the Bernoulli polynomials, the Euler polynomials, etc..

and $a_{(r)}$ denotes the Pochhammer symbol, i.e., $a_{(r)} = \frac{\Gamma(a+r)}{\Gamma(a)}$, for any integer $r > 1$, and $a_{(0)} = 1$.

It can be proved, under the additional (but natural) condition $\mathcal{P}_k^n(1) = 1$ that the sequence $\mathcal{P} = (\mathcal{P}_k^n)_{k\in\mathbb{N}}$ is an Appell sequence of monogenic polynomials, i.e., $(\mathcal{P}_k^n)' = k\mathcal{P}_{k-1}^n$. Therefore such polynomials behave like monomial functions in the sense of the complex powers $z^k = (x_0 + ix_1)^k,\ k = 1, 2, \cdots$, and allow a construction of special monogenic functions as series of the form

$$\Phi(x) = \sum_{k=0}^{\infty} a_k \mathcal{P}_k(x)\,, \tag{11}$$

with suitable chosen coefficients.

Other important properties of such sequence can also be obtained, in particular the following binomial-type formula,

$$\mathcal{P}_k^n(x) = \sum_{s=0}^{k} \binom{k}{s} x_0^{k-s} \mathcal{P}_s^n(\underline{x}) = \sum_{s=0}^{k} \binom{k}{s} c_s(n)\, x_0^{k-s}\, \underline{x}^s\,, \tag{12}$$

where

$$c_s(n) = \sum_{t=0}^{s} (-1)^t T_t^s(n)\,. \tag{13}$$

Finally, we stress the fact that it is possible to write (12) as

$$\mathcal{P}_k^n(x) = \mathcal{P}_k^n(x_0 + \omega|\underline{x}|) = u(x_0, |\underline{x}|) + \omega v(x_0, |\underline{x}|)\,, \tag{14}$$

where u and v are the real valued functions

$$u(x_0, |\underline{x}|) = \sum_{s=0}^{[\frac{k}{2}]} \binom{k}{2s} (-1)^s x_0^{k-2s} c_{2s}(n)\, |\underline{x}|^{2s}$$

and

$$v(x_0, |\underline{x}|) = \sum_{s=0}^{[\frac{k-1}{2}]} \binom{k}{2s+1} (-1)^s x_0^{k-2s-1} c_{2s+1}(n)\, |\underline{x}|^{2s+1}.$$

4.2 New Features

The package includes the functions `Tks`, `Ck` and `Pkn` to compute (10), (13) and (12), respectively.

```
Tks   gives the (k,s) element of a special triangle.
      Tks::usage = Tks[k,s] for the default dimension value.
                   Tks[k,s,n] for the n+1 dimensional case.

Ck    gives the alternating sum of Tks.
      Ck::usage = Ck[k] for the default dimension value.
                  Ck[k,n] for the n+1 dimensional case.

Pk    gives the basic monogenic polynomial of degree k.
      Pk::usage = Pk[k,x] for the default dimension value.
                  Pk[k,n,x] for the n+1 dimensional case.
```

The function `Pk` used with two arguments, the degree of the polynomial and a paravector, returns a paravector. When the dimension n increases, the corresponding output becomes very large. In such a case it is more convenient to use the alternative syntax form, where `x` is now a `ComplexLike` object of dimension `n+1` and the output is a `ComplexLike` object as in (14).

We illustrate the applicability of these functions by presenting some examples.

Example 10. A basic set of monogenic polynomials

```
In[34]:= CoordSys[x0,x1,x2,x3];
In[35]:= Tks[k,s]
```

Out[35]= $\frac{2(1+k-s)}{2+3k+k^2}$

```
In[36]:= TableForm[Table[TableForm[Table[Tks[k,s,m],{k,0,5},{s,0,k}]],
         {m,{2,3,5}}],TableDirections → Row,TableHeadings →
         {{"Tks(2)","Tks(3)","Tks(5)"}}]
```

Out[36]=

$$\begin{array}{llllll|llllll|llllll}
\multicolumn{6}{l}{\texttt{Tks(2)}} & \multicolumn{6}{l}{\texttt{Tks(3)}} & \multicolumn{6}{l}{\texttt{Tks(5)}} \\
\hline
1 &&&&&& 1 &&&&&& 1 &&&&& \\
\frac{3}{4} & \frac{1}{4} &&&&& \frac{2}{3} & \frac{1}{3} &&&&& \frac{3}{5} & \frac{2}{5} &&&& \\
\frac{5}{8} & \frac{1}{4} & \frac{1}{8} &&&& \frac{1}{2} & \frac{1}{3} & \frac{1}{6} &&&& \frac{2}{5} & \frac{2}{5} & \frac{1}{5} &&& \\
\frac{35}{64} & \frac{15}{64} & \frac{9}{64} & \frac{5}{64} &&& \frac{2}{5} & \frac{3}{10} & \frac{1}{5} & \frac{1}{10} &&& \frac{2}{7} & \frac{12}{35} & \frac{9}{35} & \frac{4}{35} && \\
\frac{63}{128} & \frac{7}{32} & \frac{9}{64} & \frac{3}{32} & \frac{7}{128} && \frac{1}{3} & \frac{4}{15} & \frac{1}{5} & \frac{2}{15} & \frac{1}{15} && \frac{3}{14} & \frac{2}{7} & \frac{9}{35} & \frac{6}{35} & \frac{1}{14} & \\
\frac{231}{512} & \frac{105}{512} & \frac{35}{256} & \frac{25}{256} & \frac{35}{512} & \frac{21}{512} & \frac{2}{7} & \frac{5}{21} & \frac{4}{21} & \frac{1}{7} & \frac{2}{21} & \frac{1}{21} & \frac{1}{6} & \frac{5}{21} & \frac{5}{21} & \frac{4}{21} & \frac{5}{42} & \frac{1}{21}
\end{array}$$

```
In[37]:= Ck[k]
```

Out[37]= $\frac{3+(-1)^k+2k}{2\left(2+3k+k^2\right)}$

```
In[38]:= TableForm[Table[Ck[k,n],{n,{2,3,5}},{k,0,10}],
         TableHeadings→{{Ck[k,2],Ck[k,3],Ck[k,5]},Range[0,10]}]
```

Out[38]=

	0	1	2	3	4	5	6	7	8	9	10
Ck[k,2]	1	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{3}{8}$	$\frac{3}{8}$	$\frac{5}{16}$	$\frac{5}{16}$	$\frac{35}{128}$	$\frac{35}{128}$	$\frac{63}{256}$	$\frac{63}{256}$
Ck[k,3]	1	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{5}$	$\frac{1}{5}$	$\frac{1}{7}$	$\frac{1}{7}$	$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{11}$	$\frac{1}{11}$
Ck[k,5]	1	$\frac{1}{5}$	$\frac{1}{5}$	$\frac{3}{35}$	$\frac{3}{35}$	$\frac{1}{21}$	$\frac{1}{21}$	$\frac{1}{33}$	$\frac{1}{33}$	$\frac{3}{143}$	$\frac{3}{143}$

```
In[39]:= TableForm[Map[Flatten,Table[{Ck[k,m],Table[Tks[k,s,m],{s,0,k}]},
         {k,0,4}]],TableHeadings → {None,{Ck[m],Tks[k,s,m]}}]
```

Out[39]=

$$\begin{array}{llllll}
\texttt{Ck[m]} & \texttt{Tks[k,s,m]} &&&& \\
1 & 1 &&&& \\
\frac{1}{m} & \frac{m+1}{2m} & \frac{m-1}{2m} &&& \\
\frac{1}{m} & \frac{m+3}{4m} & \frac{m-1}{2m} & \frac{m-1}{4m} && \\
\frac{3}{m^2+2m} & \frac{(m+3)(m+5)}{8m(m+2)} & \frac{3(m-1)(m+3)}{8m(m+2)} & \frac{3\left(m^2-1\right)}{8m(m+2)} & \frac{(m-1)(m+3)}{8m(m+2)} & \\
\frac{3}{m^2+2m} & \frac{(m+5)(m+7)}{16m(m+2)} & \frac{(m-1)(m+5)}{4m(m+2)} & \frac{3\left(m^2-1\right)}{8m(m+2)} & \frac{m^2-1}{4m(m+2)} & \frac{(m-1)(m+5)}{16m(m+2)}
\end{array}$$

```
In[40]:= TableForm[Table[{StringJoin[{"Pk[",ToString[k],",x]="}],
         Pk[k,Paravector[x0,x1,x2,x3]]},{k,0,3}]]
```

Out[40]= Pk[0,x]= 1

Pk[1,x]= Paravector $\left[\mathrm{x0}, \frac{\mathrm{x1}}{3}, \frac{\mathrm{x2}}{3}, \frac{\mathrm{x3}}{3}\right]$

Pk[2,x]= Paravector $\left[\mathrm{x0}^2 - \frac{\mathrm{x1}^2}{3} - \frac{\mathrm{x2}^2}{3} - \frac{\mathrm{x3}^2}{3}, \frac{2\mathrm{x0x1}}{3}, \frac{2\mathrm{x0x2}}{3}, \frac{2\mathrm{x0x3}}{3}\right]$

Pk[3,x]= Paravector $\left[\mathrm{x0}\left(\mathrm{x0}^2 - \mathrm{x1}^2 - \mathrm{x2}^2 - \mathrm{x3}^2\right), -\frac{1}{5}\mathrm{x1}\left(-5\mathrm{x0}^2 + \mathrm{x1}^2 + \mathrm{x2}^2 + \mathrm{x3}^2\right), -\frac{1}{5}\mathrm{x2}\left(-5\mathrm{x0}^2 + \mathrm{x1}^2 + \mathrm{x2}^2 + \mathrm{x3}^2\right), -\frac{1}{5}\mathrm{x3}\left(-5\mathrm{x0}^2 + \mathrm{x1}^2 + \mathrm{x2}^2 + \mathrm{x3}^2\right)\right]$

```
In[41]:= TableForm[Table[{StringJoin[{"Pk[",ToString[k],",2,x]="}],
         Pk[k,2,ComplexLike[x0,r]]},{k,0,3}]]
```

Out[41]= Pk[0,2,x]= ComplexLike$[1,0]$

Pk[1,2,x]= ComplexLike $\left[\mathrm{x0}, \frac{\mathrm{r}}{2}\right]$

Pk[2,2,x]= ComplexLike $\left[-\frac{\mathrm{r}^2}{2} + \mathrm{x0}^2, \mathrm{rx0}\right]$

Pk[3,2,x]= ComplexLike $\left[-\frac{3\mathrm{r}^2\mathrm{x0}}{2} + \mathrm{x0}^3, -\frac{3}{8}\left(\mathrm{r}^3 - 4\mathrm{rx0}^2\right)\right]$

4.3 Applications

In the original `Quaternions` package and, as far as we are aware, in all available quaternion packages, the elementary functions are defined by using series expansions analogue to the complex case. In fact, if a complex-valued analytic function f has a Taylor series expansion of the form

$$f(z) = f(x_0 + iy) = \sum_{k=0}^{\infty} a_k z^k \,, \tag{15}$$

the analogue $\mathbb{H}$-valued function

$$F(x) = F(x_0 + \omega|\underline{x}|) = \sum_{k=0}^{\infty} a_k x^k \tag{16}$$

is related to f by

$$F(x_0 + \omega|\underline{x}|) = \operatorname{Re}\big(f(x_0 + i|\underline{x}|)\big) + \omega \operatorname{Im}\big(f(x_0 + i|\underline{x}|)\big) \,. \tag{17}$$

Unfortunately, none of the elementary functions obtained from (16) is monogenic as the following example illustrates.

Example 11. A non-monogenic exponential function

```
In[42]:= f1=Exp[Quaternion[x0,x1,x2,x3]];
```

In[43]:= $\mathtt{Simplify}\big[\mathtt{ReplaceAll}\big[\mathtt{ToComplexLike}[\mathtt{f1}], \mathtt{x1}^2 \to \mathtt{r}^2 - \mathtt{x2}^2 - \mathtt{x3}^2\big],$
$\mathtt{Assumptions} \to \mathtt{x0} \in \mathtt{Reals}\ \&\&\ 0 < \mathtt{r} < \pi\big]$//TraditionalForm

Out[43]= $\mathrm{e}^{\mathtt{x0}}\mathtt{Cos}[\mathtt{r}] + \mathrm{e}^{\mathtt{x0}}\omega\mathtt{Sin}[\mathtt{r}]$

```
In[44]:= MonogenicQ[f1]
Out[44]= False
```

In Clifford Analysis several different methods have been developed for constructing monogenic functions as series with respect to properly chosen homogeneous monogenic polynomials. Our objective here is to obtain monogenic functions in $\mathbb{R}^{n+1}$ based on the use of the monogenic polynomials (9) instead of the (non-monogenic) complex powers of a quaternion. More precisely, for each complex-valued function f which has a Taylor expansion of the form (15), we define the monogenic analogue $\mathcal{A}_n$-valued function[5]

$$\mathcal{F}(x) = \mathcal{F}(x_0 + \omega|\underline{x}|) = \sum_{k=0}^{\infty} a_k \mathcal{P}_k^n(x) \,. \tag{18}$$

The package includes the additional function `PkSeries` which can be used to construct monogenic functions as truncated series with respect to the set (9). For certain special arguments, `PkSeries` returns the serie (18).

```
PkSeries  gives a (truncated) series expansion of a function in
          R^{n+1}, with respect to the polynomials Pkn.
PkSeries::usage =
    PkSeries[f,k,x]   gives the polynomial of order k associated
              to the Taylor expansion of the complex function f.
    PkSeries[f,Infinity,x]   gives a hypercomplex-analogue of f.
```

Example 12. Polynomial approximations

```
In[45]:= PkSeries[f,2,Paravector[x0,x1,x2,x3]]
```

[5] Since $T_s^k(n) > 0$ and $\sum_0^k T_s^k(n) = 1$, for all $n \in \mathbb{N}$, the absolute convergence of the defined function (18) is ensured because, for each $k \geq 0$, we have $|\mathcal{P}_k^n(x)| \leq \sum_{s=0}^k T_s^k(n)\ |x|^{k-s}\ |\bar{x}|^s = |x|^k$.

$$\mathtt{Out[45]=}\ \mathtt{Paravector}[f[0]+\tfrac{1}{6}(6\mathtt{x0f'}[0]+3\mathtt{x0}^2\mathtt{f''[0]}-(\mathtt{x1}^2+\mathtt{x2}^2+\mathtt{x3}^2)\mathtt{f''[0]}),\ \tfrac{1}{3}\mathtt{x1}(\mathtt{f'[0]}+\mathtt{x0f''[0]}),\tfrac{1}{3}\mathtt{x2}(\mathtt{f'[0]}+\mathtt{x0f''[0]}),\tfrac{1}{3}\mathtt{x3}(\mathtt{f'[0]}+\mathtt{x0f''[0]})]$$

```
In[46]:= PkSeries[Exp,4,ComplexLike[x0,r]]
```

$$\mathtt{Out[46]=}\ \mathtt{ComplexLike}\big[1+\mathtt{x0}+\tfrac{1}{2}(-\tfrac{\mathtt{r}^2}{3}+\mathtt{x0}^2)+\tfrac{1}{6}(-\mathtt{r}^2\mathtt{x0}+\mathtt{x0}^3)+\ \tfrac{1}{24}(\tfrac{\mathtt{r}^4}{5}-2\mathtt{r}^2\mathtt{x0}^2+\mathtt{x0}^4),\tfrac{\mathtt{r}}{3}+\tfrac{\mathtt{rx0}}{3}+\tfrac{1}{6}(-\tfrac{\mathtt{r}^3}{5}+\mathtt{rx0}^2)+\tfrac{1}{24}(-\tfrac{4\mathtt{r}^3\mathtt{x0}}{5}+\tfrac{4\mathtt{rx0}^3}{3})\big]$$

Example 13. Monogenic exponential functions in $\mathbb{R}^3$ and $\mathbb{R}^4$

```
In[47]:= CoordSys[x0,x1,x2];
In[48]:= PkSeries[Exp,Infinity,Paravector[x0,x1,x2]]
```

$$\mathtt{Out[48]=}\ \mathtt{Paravector}\left[e^{\mathtt{x0}}\mathtt{BesselJ}\left[0,\sqrt{\mathtt{x1}^2+\mathtt{x2}^2}\right],\frac{e^{\mathtt{x0}}\mathtt{x1BesselJ}\left[1,\sqrt{\mathtt{x1}^2+\mathtt{x2}^2}\right]}{\sqrt{\mathtt{x1}^2+\mathtt{x2}^2}},\frac{e^{\mathtt{x0}}\mathtt{x2BesselJ}\left[1,\sqrt{\mathtt{x1}^2+\mathtt{x2}^2}\right]}{\sqrt{\mathtt{x1}^2+\mathtt{x2}^2}}\right]$$

```
In[49]:= CoordSys[4];
In[50]:= PkSeries[Exp,Infinity,ComplexLike[x0,r]]
```

$$\mathtt{Out[50]=}\ \mathtt{ComplexLike}\left[\frac{e^{\mathtt{x0}}\mathtt{Sin[r]}}{\mathtt{r}},\frac{e^{\mathtt{x0}}\left(-\mathtt{Cos[r]}+\frac{\mathtt{Sin[r]}}{\mathtt{r}}\right)}{\mathtt{r}}\right]$$

5 Final Remarks

This paper presents briefly a collection of functions to endow the Mathematica standard package `Quaternions` with new functionalities in the framework of Clifford Analysis. It should be considered work in progress in its present form. Future work on the package will include, for example, implementation of different techniques to generate monogenic functions and more applications of the Appell set (9) (special functions, orthogonal polynomials, etc.).

Acknowledgments This research was partially supported by the Research Centre of Mathematics of the University of Minho and by the Center for Research and Development in Mathematics and Applications of the University of Aveiro, both through the Portuguese Foundation for Science and Technology Pluriannual Funding Program.

References

1. Abłamowicz, R.: Computations with Clifford and Graßmann algebras. Adv. Appl. Clifford Algebr. 19(3-4), 499–545 (2009)
2. Abłamowicz, R., Baylis, W.E., Branson, T., Lounesto, P., Porteous, I., Ryan, J., Selig, J.M., Sobczyk, G.: Lectures on Clifford (geometric) algebras and applications. Birkhäuser Boston Inc., Boston, MA (2004), edited by Abłamowicz and Sobczyk
3. Abłamowicz, R., Fauser, B.: Mathematics of Clifford – a Maple package for Clifford and Graßmann algebras. Adv. Appl. Clifford Algebr. 15(2), 157–181 (2005)
4. Altmann, S.L.: Rotations, quaternions, and double groups. Oxford Science Publications, The Clarendon Press Oxford University Press, New York (1986)
5. Bock, S., Gürlebeck, K.: On a generalized Appell system and monogenic power series. Math. Methods Appl. Sci. 33(4), 394–411 (2010)
6. Brackx, F., Delanghe, R., Sommen, F.: Clifford analysis. Pitman, Boston-London-Melbourne (1982)
7. Cação, I., Malonek, H.: On complete sets of hypercomplex Appell polynomials. In: Simos, T.E., Psihoyios, G., Tsitouras, C. (eds.) AIP Conference Proceedings. vol. 1048, pp. 647–650 (2008)
8. Clifford, P.: Applications of Grassmann's Extensive Algebra. Amer. J. Math. 1(4), 350–358 (1878)
9. Falcão, M.I., Cruz, J., Malonek, H.R.: Remarks on the generation of monogenic functions. 17th Inter. Conf. on the Appl. of Computer Science and Mathematics on Architecture and Civil Engineering, Weimar (2006)

10. Falcão, M.I., Malonek, H.R.: Generalized exponentials through Appell sets in $\mathbb{R}^{n+1}$ and Bessel functions. In: Simos, T.E., Psihoyios, G., Tsitouras, C. (eds.) AIP Conference Proceedings. vol. 936, pp. 738–741 (2007)
11. Fueter, R.: Die Funktionentheorie der Differetialgleichungen $\Delta u = 0$ und $\Delta\Delta u = 0$ mit vier reellen Variablen. Comm. Math. Helv. (7), 307–330 (1934-35)
12. Fueter, R.: Über die analytische Darstellung der regulären Funktionen einer Quaternionenvariablen. Comment. Math. Helv. 8(1), 371–378 (1935)
13. Gürlebeck, K., Malonek, H.: A hypercomplex derivative of monogenic functions in $\mathbb{R}^{n+1}$ and its applications. Complex Variables Theory Appl. 39, 199–228 (1999)
14. Gürlebeck, K., Sprössig, W.: Quaternionic and Cliford calculus for physicists and engineers. John Wiley & Sons (1997)
15. Gürlebeck, K., Habetha, K., Sprößig, W.: Holomorphic functions in the plane and n-dimensional space. Birkhäuser Verlag, Basel (2008)
16. Harder, D.W.: Quaternions in Maple. Kotsireas, Ilias S. (ed.), Maple conference 2005. Proceedings of the conference, Waterloo Ontario, Canada, July 17–21 (2005)
17. Lávička, R.: Canonical bases for sl(2,c)-modules of spherical monogenics in dimension 3. Arch. Math. (Brno) 46(5), 339–349 (2010)
18. Malonek, H.: A new hypercomplex structure of the euclidean space $\mathbb{R}^{m+1}$ and the concept of hypercomplex differentiability. Complex Variables 14, 25–33 (1990)
19. Malonek, H.: Power series representation for monogenic functions in $\mathbb{R}^{n+1}$ based on a permutational product. Complex Variables, Theory Appl. 15, 181–191 (1990)
20. Malonek, H.R.: Quaternions in applied sciences. a historical perspective of a mathematical concept. 17th Inter. Conf. on the Appl. of Computer Science and Mathematics on Architecture and Civil Engineering, Weimar (2003)
21. Malonek, H.R., Falcão, M.I.: Special monogenic polynomials—properties and applications. In: Simos, T.E., Psihoyios, G., Tsitouras, C. (eds.) AIP Conference Proceedings. vol. 936, pp. 764–767 (2007)
22. Sangwine, J., Le Bihan, N.: Quaternion Toolbox for Matlab. `http://qtfm.sourceforge.net` (2005)
23. Sudbery, A.: Quaternionic analysis. Math. Proc. Camb. Phil. Soc. 85, 199–225 (1979)