Clifford for *Mathematica.*

A *Mathematica* package for doing Clifford Algebra. Version 1.2

By

GERARDO ARAGÓN-CAMARASA

University of Glasgow, Department of Computing Science.

JOSÉ LUIS ARAGÓN

Universidad Nacional Autónoma de México, Centro de Física Aplicada y Tecnología Avanzada.

GERARDO ARAGÓN GONZÁLEZ

Universidad Nacional Autónoma Metropolitana, Unidad Acapotzalco, Departamento de Energía.

M. ANTONIO RODRÍGUEZ-ANDRADE

Instituto Politécnico Nacional, Escuela Superior de Física y Matemáticas.

1. Installation.

clifford.m

In order to install clifford.m (core of the Clifford Algebra package), just copy the `clifford.m` file into the directory;

*(Mathematica installation dir)***\Wolfram Research\Mathematica\X.X\AddOns\ExtraPackages**

Clifford.nb

To have access to a palette that contains some functions of clifford.m without typing the whole word. Just copy the `Clifford.nb` file to;

*(Mathematica installation dir)***\Wolfram Research\Mathematica\X.X\SystemFiles\FrontEnd\Palettes**

Now, the palette will be in the Palettes submenu of the File menu.

Documentation.

The documentation includes this user guide and a brief description of all the functions of `clifford.m`. Thus, to put the documentation in the help browser of *Mathematica's* FrontEnd, just copy the Clifford.m folder into:

*(Mathematica installation dir)***Wolfram Research\Mathematica\X.X\Documentation\English\AddOns**

Now, in order to view it in the Help Browser, it must be edited the `BrowserCategories.m` of the last folder with the next lines;

```
HelpDirectoryListing[{ToFileName[{"AddOns","Clifford"}]},False],
Item[Delimiter],
```

So, the `BrowserCategories.m` would be like the following example.

```
BrowserCategory["Add-ons & Links", None,
{

HelpDirectoryListing[{ToFileName[{"AddOns", "Clifford"}]}, False],
Item[Delimiter],

BrowserCategory["Wolfram Research Products", None,
     {
      BrowserCategory["Mathematica Applications", None,
         {
            Item["Mathematica Applications from Wolfram Research", ...,
            Item[Delimiter],
            Item["Mathematica Packages from Independent Developers", ...,
            Item["Other Mathematica Applications", ...
            }],
     BrowserCategory["Wolfram Education Training", None,
        {...
...
```

Finally, go to the `Help` menu and select `Rebuild Help Index`. Now, the help for clifford.m is in the `Help Browser`.

2. Introduction.

The Clifford algebra of the vector space $R^{p,q}$, with a bilinear form $\langle x, y\rangle$ of signature p $_q$and an orthonormal basis $\{e_1, e_2, \ldots, e_n\}$, $i = 1, 2, \ldots, n\ (= p + q)$, is generated by $R^{p,q}$ with the relation

$$e_i\, e_j + e_j\, e_i = 2 \left\langle e_i, e_j \right\rangle$$

where

$$\begin{aligned} &\left\langle e_i, e_j\right\rangle = 0 \ \text{ if } \ i = j \\ &\langle e_i, e_i\rangle = 1 \ \text{ if } \ i = 1, \ \ldots, p \\ &\langle e_i, e_i\rangle = -1 \ \text{ if } \ i = p+1, \ \ldots, n \end{aligned}$$

`Clifford.m` is a package for doing general calculations with Clifford Algebra of $R^{p,q}$, using *Mathematica* 5.0 or higher. All results are given in terms of the orthonormal basis vectors $\{e_1, e_2, \ldots, e_n\}$.

In session with `Clifford.m`, basis vectors e_i are denoted by `e[i]`. For instance, the multivectors

$$A = a\, e_1 + b\, e_2$$
$$B = 1 + a \times b + (5 - a)\, e_1\, e_2$$
$$T = 17\, e_1 + a^2\, e_1\, e_2\, e_3,$$

must be written as

```
<< "clifford.m"
```

```
A = a * e[1] + b * e[2];
B = 1 + a * b + (5 - a) * e[1] * e[2];
T = 17 * e[1] + a^2 * e[1] * e[2] * e[3];
```

Care must be taken in preserving the canonical order of the expression since we are using the commutative product * of *Mathematica* and expression are automatically rewritten in canonical order. The use of the function `GeometricProduct` is recommended in order to avoid mistakes, thus for example the multivector $B = e_1\, e_3\, e_2$ must be written as

```
B = GeometricProduct[e[1], e[3], e[2]]
```

```
-e1 e2 e3
```

But as a short cut we can type directly

```
B = -e[1] e[2] e[3]
```

```
-e1 e2 e3
```

The signature of the bilinear form $\langle x, y \rangle$ can be set by using `$SetSignature=p`. If no value is specified at the beginning of the session, the default is $p = 20$.

Whit the exception of the function Dual, it is not necessary to define the dimension of the vector space $R^{p,q}$. Given two or more multivectors, the maximum dimension of the space where they are embedded is calculated automatically.

3. Listing of implicit functions.

2.1 `Coeff[m,b]`

Description: Extracts the coefficient of the blade `b` in the multivector `m`.

Arguments: `b` is a blade of grade and `m` is a multivector.

2.2 `Dual[m,d]`

Description: Calculates the dual of the multivector `m` in R^d.

Arguments: `m` is a multivector and `d` is a positive integer.

2.3 `e[i]`

Description: `e[i]` is used to denote the i-th basis vector of R^d.

Arguments: `i` is a integer greater than zero.

2.4 `GADraw[m,v]`

Description: Plots a multivector `m` in R^3. To change the plot's view `v`, it must be used the `ViewPoint` function.

Arguments: `m` is a multivector and `v` is the view point of the plot.

Comments: `v` can be omitted and the default value is `ViewPoint→{0,1,0}`.

2.5 `GeometricCos[m,n]`

Description: Calculates the power series of the function `Cos` of the multivector `m` to a power `n`.

Arguments: `m` is a multivector and `n` a positive integer.

Comments: `n` can be omitted and the default value is 10.

2.6 `GeometricExp[m,n]`

Description: Calculates the power series of the function `Exp` of the multivector `m` to a power `n`.

Arguments: `m` is a multivector and `n` a positive integer.

Comments: `n` can be omitted and the default value is 10.

2.7 `GeometricPower[m,n]`

Description: Calculates the n-th power of the multivector `m`.

Arguments: `m` is a multivector and `n` a positive integer.

2.8 `GeometricProduct[m1,m2,...]`

Description: Calculates the geometric product of the multivectors `m1,m2,...`

Arguments: `m1,m2,...` are multivectors.

2.9 `GeometricProductSeries[sym,m,n]`

Description: Calculates the power series of the function `sym` of the multivector `m` to a power `n`.

Arguments: `sym` is a *Mathematica* function, `m` is a multivector and `n` a positive integer.

Comments: `sym` is any function which can be represented as a power series about zero. `n` can be omitted and the default value is 10.

2.10 `GeometricSin[m,n]`

Description: Calculates the power series of the function `Sin` of the multivector `m` to a power `n`.

Arguments: `m` is a multivector and `n` a positive integer.

Comments: `n` can be omitted and the default value is 10.

2.11 `GeometricTan[m,n]`

Description: Calculates the power series of the function `Tan` of the multivector `m` to a power `n`.

Arguments: `m` is a multivector and `n` a positive integer.

Comments: `n` can be omitted and the default value is 10.

2.12 `Grade[m,r]`

Description: Extracts the term of grade `r` from the multivector `m`.

Arguments: `m` is a multivector and `r` a positive integer.

2.13 `i`

Description: Denotes the first complex component of a quaternion (see also `j` and `k`)

Arguments: None.

2.14 `Im[q]`

Description: Extracts the complex part of a quaternion `q`.

Arguments: `q` is a quaternion.

2.15 `InnerProduct[m1,m2,...]`

Description: Calculates the inner product of the multivectors `m1,m2,...`

Arguments: `m1,m2,...` are multivectors.

2.16 `j`

Description: Denotes the second complex component of a quaternion (see also `i` and `k`)

Arguments: None.

2.17 `k`

Description: Denotes the third complex component of a quaternion (see also `i` and `j`)

Arguments: None.

2.18 `Magnitude[m]`

Description: Calculates the magnitude of the multivector `m`.

Arguments: `m` is a multivector.

2.19 `MutivectorInverse[m]`

Description: Calculates (if it exists) the inverse of the multivector `m`.

Arguments: `m` is a multivector.

2.20 `OuterProduct[m1,m2,...]`

Description: Calculates the outer product of the multivectors `m1,m2,...`

Arguments: `m1,m2,...` are multivectors.

2.21 `Projection[v,b]`

Description: Projects the vector `v` onto the space spanned by the blade `b`.

Arguments: `v` is a vector and `b` a r-blade.

2.22 `Pseudoscalar[n]`

Description: Gives the pseudoscalar (volume element) of R^n.

Arguments: `n` is a positive integer.

2.23 `QuaternionConjugate[q]`

Description: Calculates the conjugate of the quaternion `q`.

Arguments: `q` is a quaternion.

2.24 `QuaternionInverse[q]`

Description: Calculates the inverse of the quaternion `q`.

Arguments: `q` is a quaternion.

2.25 `QuaternionMagnitude[q]`

Description: Calculates the magnitude of the quaternion `q`.

Arguments: `q` is a quaternion.

2.26 `QuaternionProduct[q1,q2,...]`

Description: Calculates the product of the quaternions `q1,q2,...`

Arguments: `q1,q2,...` are quaternions.

2.27 `Re[q]`

Description: Extracts the real part of the quaternion `q`.

Arguments: `q` is a quaternion.

2.28 `Reflection[v,w,x]`

Description: Calculates the specular reflection of the vector `v` by the plane spanned by the vectors `w` and `x`.

Arguments: `v`, `w` and `x` are vectors.

2.29 `Rejection[v,b]`

Description: Calculates the orthogonal projection of the vector `v` onto the orthogonal complement to the space spanned by the blade `b`.

Arguments: `v` is a vector and `b` a r-blade.

2.30 `Rotation[v,w,x,theta]`

Description: Rotates the vector `v`, by an angle `theta`. The plane spanned by `w` and `x` is left invariant.

Arguments: `v`, `w` and `x` are vectors and `theta` is the rotation angle in degrees.

Comments: `theta` can be omitted and in such case, the rotation angle is that formed by the vectors `w` and `x`.

2.31 `ToBasis[v]`

Description: Transforms a vector from the *Mathematica* notation (list) to a linear combination of vectors `e[i]`.

Arguments: `v` is a vector given in standard notation (list).

2.32 `ToVector[v,d]`

Description: Transforms a vector from a linear combination of vectors or multivectors in the canonical form `e[i]` to the standard notation in *Mathematica* (d-dimensional list).

Arguments: `v` is a vector and `d` positive integer.

Comments: `d` can be omitted and in such case the list's dimension is the greatest dimension of the basis vectors `e[i]`.

2.33 `Turn[m]`

Description: Gives the reverse of the multivector `m`.

Arguments: `m` is a multivector.

4. Simple examples.

This loads the package.

```
<< "clifford.m"
```

Here are 3 multivectors.

```
u = a + 3 * e[1] + b * e[3];
v = a * e[1] * e[2] * e[3];
w = e[2] + b * e[1] * e[2] * e[3] * e[4];
```

The geometric product uvw is:

```
B = GeometricProduct[u, v, w]
```

$a\,b\,e_1 - 3\,a\,e_3 - a^2\,e_1\,e_3 - a^2\,b\,e_4 - 3\,a\,b\,e_1\,e_4 - a\,b^2\,e_3\,e_4$

The outer (wedge) product between u and v is:

```
OuterProduct[u, v]
```

$a^2\,e_1\,e_2\,e_3$

The operation $\langle \tilde{w} \rangle_4$ is:

```
Grade[Turn[w], 4]
```

$b\,e_1\,e_2\,e_3\,e_4$

The multivector $e^{1+e_1 e_2}$ is:

```
Simplify[GeometricExp[1 + e[1] * e[2], 5]]
```

$\frac{1}{30}\left(44 + 69\,e_1\,e_2\right)$

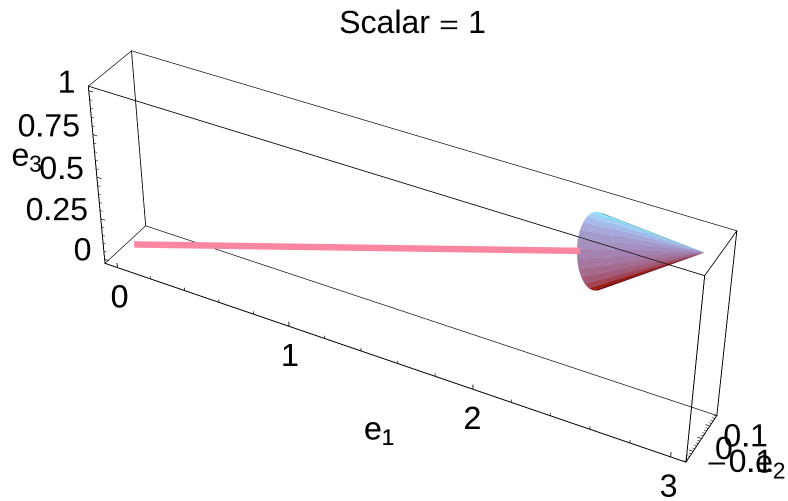
The product between the quaternions $(2 + \mathbf{i} + 3\,\mathbf{k})\,(a + \mathbf{k})$ is:

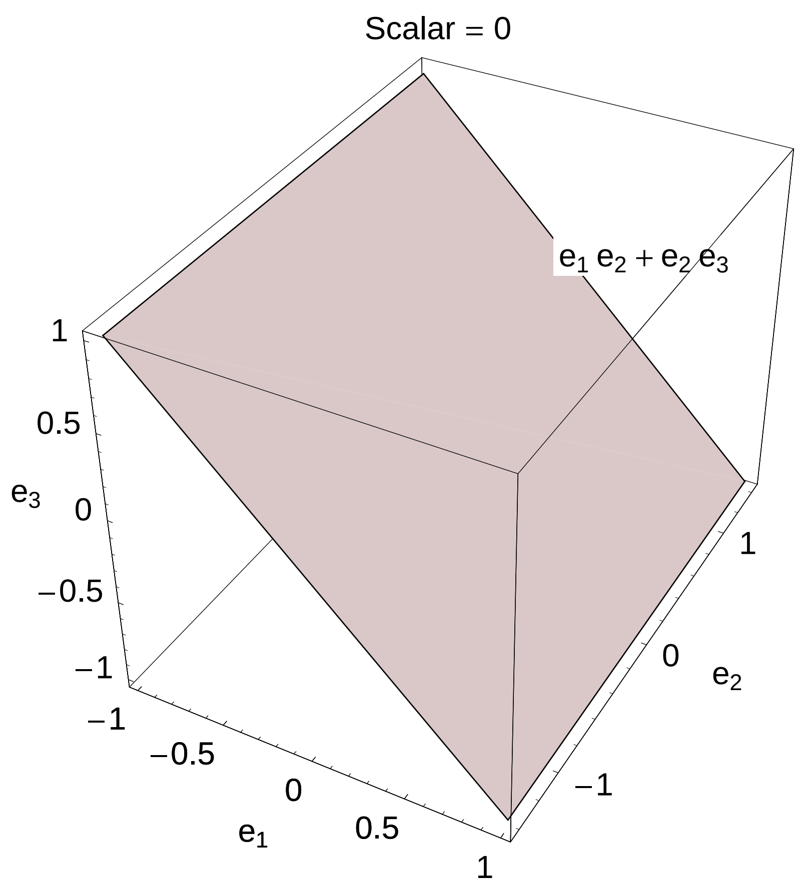
```
QuaternionProduct[2 + i + 3 * k, a + k]
```

$-3 + 2\,a + a\,i - j + 2\,k + 3\,a\,k$

The plot of u and the plane e[1]e[2]+e[2]e[3] (the vectors must be numeric in order to plot) is:

```
a = 1;
b = 1;
plot1 = GADraw[u];
plot2 = GADraw[e[1] e[2] + e[2] e[3]];
```





Finally, in order to put together `u` and the plane `e[1]e[2]` it is used the `Show` function:

```
Show[plot1, plot2];
```