DEPARTMENT OF MATHEMATICS
TECHNICAL REPORT

COMPUTATIONS WITH CLIFFORD AND GRASSMANN ALGEBRAS

RAFAL ABLAMOWICZ

MAY 2009

No. 2009-4

TENNESSEE TECHNOLOGICAL UNIVERSITY
Cookeville, TN 38505

Computations with Clifford and Grassmann Algebras

Rafal Ablamowic (B)

endowed with an arbitrary bilinear
smann basis in $C\ell(Q)$ but when
it can also compute in a dotted
putations are discussed.

15A66, 68W30.

rmal group, contraction, dotted
n algebra, Hopf algebra, multivec-
ular value decomposition, spinors,

s

1. Introduction

Some twenty years ago, late Professor Pertti Lounesto together with his colleagues at Helsinki University of Technology developed CLICAL, a first semi-symbolic \Clifford algebra calculator". [32] Along with it, Pertti brought to the world of Clifford algebraists a concept of experimental mathematics, algorithmic understanding, and counter examples

In CLIFFORD these basis monomials are written as strings f Id, e1, . . . , e9, e1we2, e1we3, . . . , e1we2we3, . . . g although they can be aliased to f Id, e1, . . . , e9, e12, e13, . . . , e123, . . . g to shorten input. Here e1we2 is a string that denotes $e_1 \wedge e_2$ and Id denotes the identity 1 in ^{V} V: However, CLIFFORD can also use one-character long symbolic indices as in eiwej which stands for $e_i \wedge e_j$:

$$\mathrm{eiwejwek} + K_{j;k}\;\mathrm{ei}\quad K_{i;k}\;\mathrm{ej} + K_{i;j}\;\mathrm{ek}$$

The form B can be numeric or symbolic. For example, when

```
> B:=matrix(2,2,[1,a,a,1]);
```

$$B := \begin{bmatrix} 1 & a \\ a & 1 \end{bmatrix}$$

then the Grassmann basis for $C\ell(B)$ or $\bigwedge V$ will be:

```
> cbas:=cbasis(2);
```

$$\mathit{cbas} := [\mathit{Id}, \mathit{e1}, \mathit{e2}, \mathit{e12}]$$

while the Clifford multiplication table of the basis Grassmann monomials will look as follows:

```
> MultTable:=matrix(4,4,(i,j)->cmul(cbas[i],cbas[j]))          ;
```

$$\mathit{MultTable} := \begin{bmatrix} \mathit{Id} & \mathit{e1} & \mathit{e2} & \mathit{e12} \\ \mathit{e1} & \mathit{Id} & \mathit{e12} + a\,\mathit{Id} & \mathit{e2} \quad a\,\mathit{e1} \\ \mathit{e2} & \mathit{e12} + a\,\mathit{Id} & \mathit{Id} & a\,\mathit{e2} \quad \mathit{e1} \\ \mathit{e12} & a\,\mathit{e1} \quad \mathit{e2} & \mathit{e1} \quad a\,\mathit{e2} & (\ 1 + a^2)\,\mathit{Id} \end{bmatrix}$$

Irrespective of the bilinear form chosen, the Grassmann multiplication table will always remain as:

```
> wedgetable:=matrix(4,4,(i,j)->wedge(cbas[i],cbas[j]          ));
```

$$\mathit{wedgetable} := \begin{bmatrix} \mathit{Id} & \mathit{e1} & \mathit{e2} & \mathit{e12} \\ \mathit{e1} & 0 & \mathit{e12} & 0 \\ \mathit{e2} & \mathit{e12} & 0 & 0 \\ \mathit{e12} & 0 & 0 & 0 \end{bmatrix}$$

Let $B = g + F$ where g and F are, respectively, the symmetric and the anti70.0733(,)-372.645(t)3.44941(h)13e a*[(7)-10.

7

7a
7

Then, the Clifford multiplication table of the basis monomials in $C\ell(B)$ will be as follows:

```
> MultTable:=matrix(4,4,(i,j)->cmul(cbas[i],cbas[j]))          ;
```

algebras $C\ell(Q)$; $1 \le n = p+q \le 9$; and for any signature (p, q) has been precomputed [3] and can be retrieved from CLIFFORD with a procedure matKrepr: For example, 1-vectors e_1 and e_2 in $C\ell_2$ have the following spinor representation in the basis $\{f, e2\,\&c\,f\}$ of $S = C\ell_2 f$:[2]

```
> matKrepr([2,0]);
```

$$[e1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, e2 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}]$$

In another example, Clifford algebra $C\ell_3$ of $\mathbb{R}^3$ is isomorphic with $\mathrm{Mat}(2, \mathbb{C})$:

```
> B:=linalg[diag](1,1,1):clidata([3,0]);
```

$$[complex, 2, simple, \frac{1}{2}Id + \frac{1}{2}e1, [Id, e2, e3, e23], [Id, e23], [Id, e2]]$$

and its spinor representation is given in terms of Pauli matrices:

```
> matKrepr([3,0]);
```

$$[e1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, e2 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, e3 = \begin{bmatrix} 0 & e23 \\ e23 & 0 \end{bmatrix}]$$

Notice that $F = \mathrm{span}\{Id, e23\}$ ($e23 = e2we3$) is a subalgebra of $C\ell_3$ isomorphic to

eis a 4

```
> `M1 &cm M1` = evalm(M1 &cm M1),`M2 &cm M2` = evalm(M2 &cm M2),
> `M3 &cm M3` = evalm(M3 &cm M3);
> `e1 &c e1` = e1 &c e1,`e2 &c e2` = e2 &c e2,`e3 &c e3` = e3 &c e3;
```

$$M1\ \&cm\ M1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix};\ M2\ \&cm\ M2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix};\ M3\ \&cm\ M3 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

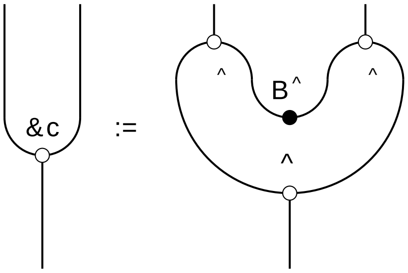
$$e1\ \&c\ e1 = Id;\ e2\ \&c\ e2 = Id;\ e3\ \&c\ e3 = Id$$

The procedure matKrepr gives the linear isomorphism C`([(()3&4M r er

Clifford algebras in higher dimensions. The BIGEBRA package is described in [10]. For more information about any CLIFFORD or BIGEBRA procedure, type ?Clifford or ?Bigebra and the second recursion of the process gives now

$$= \quad B$$

The procedure cmulRS is encoded a non-recursive Rota-Stein cliffordization. See [10, 20, 22, 24, 40] and BIGEBRA help pages for additional references[8]. The cliffordization process is based on the Hopf algebra theory. The Clifford product is obtained from the Grassmann wedge product and its Grassmann co-product as shown by the following tangle:



(4)

Here ^ is the Grassmann exterior wedge product and ^ is the Grassmann exterior co-product which is obtained from the wedge product by a categorial duality: To every algebra over a linear space A with a product we find a co-algebra with a co-product over the same space by reversing all arrows in all axiomatic commutative diagrams. Note that the co-product splits each input `factor'

$$\begin{tikzcd}
C\ell(B)_{\wedge} \quad C\ell(B)_{\wedge} \arrow[r, "1 \quad (\ldots)_F"] \arrow[d, "\rfloor_B"] & C\ell(B)_{\wedge} \quad C\ell(B)_{\underline{\wedge}} \arrow[d, "\rfloor_B"] \\
C\ell(B)_{\wedge} & C\ell(B)_{\underline{\wedge}} \arrow[l, "(\ldots)\ _F"']
\end{tikzcd}$$

Diagram 2. Contraction w.r.t. wedge and dotted wedge.

true

$$\begin{tikzcd}
C`(g)_{\wedge} \quad C`(g)_{\wedge} \arrow[r, "(:::)_F \quad (:::)_F"] \arrow[d, "cmul[g]"] & C`(g)_{\underline{\wedge}} \quad C`(g)_{\underline{\wedge}} \arrow[d, "cmul[B]"] \\
C`(g)_{\wedge} & C`(g)_{\underline{\wedge}} \arrow[l, "(:::)\ _F"']
\end{tikzcd}$$

Diagram 3. Clifford multiplications cmul[g] and cmul[B] w.r.t. dotted and undotted basis.

```
> uv:=cmulg(u,v): #Clifford product w.r.t. g in Cl(g) in wedg         e basis
```

Now, we convert u and v to u_F and v_F ;

$$\begin{tikzcd}
C\ell(B)_{\wedge} \arrow[r, "reversion\ [B](: : :)"] \arrow[d, "reversion\ [g](: : :)"'] & C\ell(B)_{\dot\wedge} \\
C\ell(B)_{\wedge} \arrow[ru, "(: : :)_{2F}"'] &
\end{tikzcd}$$

Diagram 6. Relation between reversion[g] and reversion[B] and the basis transformation $(: : :)_{2F}$:

We illustrate how the various reversions are related in the following commutative diagram:

The reader should note that the map, depicted by the diagonal arrow in Diagram 6, involves a change of basis induced by the antisymmetric bilinear form $2F$ and not F: The factor 2 is crucial and appears due to an asymmetry between the undotted and dotted bases. This suggests to introduce a symmetrically related triple of bases w.r.t. $\frac{1}{2}F; F \quad 0$ and $\frac{1}{2}F$: In such a setting, F (resp. F) connects the two dotted bases induced by $\frac{1}{2}F$:

Observe in the pre-last display above that only when $B_{1;2} = B_{2;1}$; the result $e_1 \wedge e_2$ known from the theory of classical Clifford algebras is obtained. Likewise,

```
> cbas:=cbasis(3);
```

```
          cbas := [ Id ; e1; e2; e3; e1we2; e1we3; e2we3; e1we2we3]
```

```
> map(reversion,cbas,B);
```

```
[Id ; e1; e2; e3;   e1we2   2 F1; 2 Id ;   e1we3   2 F1; 3 Id ;   e2we3   2 F2; 3 Id ;
                                       2 F2; 3 e1 + 2 F1; 3 e2   2 F1; 2 e3   e1we2we3]
```

If instead of B we use a symmetric matrix $g = g^T$

7. Spinor Representation of $C\ell(Q)$ in Minimal Left Ideals

See [3] for a complete treatment of symbolic computation of spinor representations of simple and semisimple Clifford algebras. Here we provide some basic facts and a few examples. We will use a procedure spinorKrepr from CLIFFORD:

Procedure spinorKrepr finds a matrix spinor representation of any Clifford polynomial in a minimal left ideal S

```
> dim:=3:B:=linalg[diag](1,1,1):#define the bilinear form B for Cl(3,0)
> clibasis:=cbasis(dim): #compute Clifford basis for Cl(3,0)
> data:=clidata(B); #retrieve and display data about Cl(3,0)
```

$$data := [\,complex;\ 2;\ simple;\ \frac{Id}{2} + \frac{e1}{2};\ [Id\,;\ e2;\ e3;\ e23];\ [Id\,;\ e23];\ [Id\,;\ e2]]$$

```
> f:=data[4]: #assign pre-stored idempotent to f or use your own here
> sbasis:=minimalideal(clibasis,f,'left'):#compute a real basis in Cl(3,0)f
> Kbasis:=Kfield(sbasis,f); #compute a basis for the field K
```

$$Kbasis := [[\,\frac{Id}{2} + \frac{e1}{2};\ \frac{e23}{2} + \frac{e123}{2}];\ [Id\,;\ e23]]$$

```
> SBgens:=sbasis[2]: #generators for a real basis in S
> FBgens:=Kbasis[2]; #generators for K are two since K=C
```

$$FBgens := [\,Id\,;\ e23]$$

```
> K_basis:=spinorKbasis(SBgens,f,FBgens,'left');
```

$$K\_basis := [[\,\frac{Id}{2} + \frac{e1}{2};\ \frac{e2}{2} \quad \frac{e12}{2}];\ [Id\,;\ e2];\ left\,]$$

Here are the matrices representing 1-vector basis monomials of C`$_{3;0}$: Matrices sigma[1]; sigma[2] and sigma[3] are the well-known Pauli matrices with entries in the field K :

```
> sigma[1],sigma[2],sigma[3]:=
> op(map(spinorKrepr,[e1,e2,e3],K_basis[1],FBgens,'left'));
```

$$_1;\ \ _2;\ \ _3 := \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix};\ \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix};\ \begin{bmatrix} 0 & e23 \\ e23 & 0 \end{bmatrix}$$

```
> s:=f1 &c psi[1] + f2 &c psi[2];#remember that S is a right K-ve      ctor space
```

$$s := \frac{a\,Id}{2} + \frac{be23}{2} + \frac{a\,e1}{2} + \frac{be123}{2} \quad \frac{c\,e12}{2} \quad \frac{d\,e13}{2} + \frac{c\,e2}{2} + \frac{d\,e3}{2}$$

```
>
```

```
> B:=diag(1,1,1); #define B for Cl(3,0)
```

$$B := \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

```
> dim:=coldim(B):eval(makealiases(dim)):
> data:=clidata(B); #retrieve and display data about Cl(B)
```

$$data := [\,complex;\ 2;\ simple;\ \frac{Id}{2} + \frac{e1}{2};\ [Id\,;\ e2;\ e3;\ e23];\ [Id\,;\ e23];\ [Id\,;\ e2]]$$

```
> f:=data[4]: #assign pre-stored idempotent to f or use your own here
> for i from 1 to nops(data[7]) do f||i:=data[7][i] &c f od;
```

$$f1 := \frac{Id}{2} + \frac{e1}{2};\quad f2 := \frac{e2}{2} \quad \frac{e12}{2}$$

```
> Kbasis:=data[6]; #here K = C
```

$$Kbasis := [\,Id\,;\ e23]$$

Let's define arbitrary (complex) spinor coefficients psi1 ; psi2 ; phi1 and phi2 for two spinors and in $S = C`_{3;0}f$ ' $C^2$: Notice, that these coefficients belong to a subalgebra K of $C`_{3;0}$ spanned by f 1; e_{23}g that is isomorphic to C since $e_{23}^2 =$ 1: Recall also that the left minimal ideal $S = C`(Q)f$ is a right K -module. That's why the 'complex' coefficients must be written on the right of the spinor basis elements f1 and f2 in S:

```
> psi1:=psi11 * Id + psi12 * e23;psi2:=psi21 * Id + psi22 * e23;
```

$$1 := \ 11\,Id + \ 12\,e23;\qquad 2 := \ 21\,Id + \ 22\,e23$$

```
> phi1:=phi11 * Id + phi12 * e23;phi2:=phi21 * Id + phi22 * e23;
```

$$1 := \ 11\,Id + \ 12\,e23;\qquad 2 := \ 21\,Id + \ 22\,e23$$

Thus, $= f_1\ _1 + f_2\ _2$ and $= f_1\ _1 + f_2\ _2$ which is shown in Maple with a help of an unevaluated Clifford product climul as follows:

```
> psi:='f1 &c psi1' + 'f2 &c psi2';phi:='f1 &c phi1' + 'f2 &c phi2';
```

$$:= climul(\,f1\,;\ \ 1) + climul(\,f2\,;\ \ 2);\qquad := climul(\,f1\,;\ \ 1) + climul(\,f2\,;\ \ 2)$$

Now, we compute $_+(\ ;\ )$ while we store the purespinor u under the name purespinor1 : Notice, that $_+$ is invariant under the unitary group U(2):

```
> beta_plus(psi,phi,f,'purespinor1');purespinor1;
```

$$(\ 22\ \ 22 + \ 21\ \ 21 + \ 11\ \ 11 + \ 12\ \ 12)\,Id$$
$$+\,(\ 21\ \ 22\ \ \ 12\ \ 11 + \ 11\ \ 12\ \ \ 22\ \ 21)\,e23$$
$$Id$$

Observe that $_{+}$ (;

We will show how to find continuous families of idempotents in a Clifford algebra $C\ell(Q)$ by finding a general solution to the equation $f^2 = f$ with a procedure clisolve. As low dimensional examples, we will use $C\ell_{2,0}$, $C\ell_{1,1}$ and $C\ell_{3,0}$.

Example 4. Families of idempotents in $C\ell$

```
> f:=add(x[i]*bas[i],i=1..2^dim_V);
```

$$f := x_1\, Id + x_2\, e1 + x_3\, e2 + x_4\, e12$$

```
> sol:=map(allvalues,clisolve(cmul(f,f)-f,f)):sol_real:=remove(has,sol,I);
```

$$\begin{aligned} sol\_real := [0,\ Id,\ & \frac{Id}{2} + \frac{1}{2}\sqrt{1 - 4x_4^2}\, e2 + x_4\, e12,\ \frac{Id}{2} - \frac{1}{2}\sqrt{1 - 4x_4^2}\, e2 + x_4\, e12,\\ & \frac{Id}{2} + \frac{1}{2}\sqrt{1 + 4x_3^2 - 4x_4^2}\, e1 + x_3\, e2 + x_4\, e12,\\ & \frac{Id}{2} - \frac{1}{2}\sqrt{1 + 4x_3^2 - 4x_4^2}\, e1 + x_3\, e2 + x_4\, e12] \end{aligned}$$

```
> map(x -> is(simplify(cmul(x,x)=x)),sol_real);
```

$$[true,\ true,\ true,\ true,\ true,\ true]$$

Thus, like in the Euclidean case, we find that

$$\frac{1}{2} \pm \frac{1}{2}\sqrt{1 + 4x_3^2 - 4x_4^2}\, e_1 + x_3 e_2 + x_4 e_1 \wedge e_2 \tag{17}$$

gives a two parameter family of idempotents provided $1 + 4x_3^2 - 4x_4^2 \geq 0$. Like in the Euclidean case we find that the idempotents in the pair (17) do not add up to 1 and do not mutually annihilate unless $x_3 = x_4 = 0$. In that case we find graded idempotents $\frac{1}{2} \pm \frac{1}{2} e_1 \wedge e_2$.

In the anti-Euclidean signature $(0, 2)$ we only find, as expected, trivial idempotents in $C\ell_{0,2} \simeq \mathbb{H}$. In higher dimensions, for example in $C\ell_{3,0}$, one also finds families parameterized by more than two parameters.

10. Vahlen Matrices

For the background material on Vahlen matrices and conformal transformations, see [15, 31, 33, 34, 38]. Procedure isVahlenmatrix determines if a given 2×2 Clifford matrix $V \in Mat(2, C\ell(Q))$ is a Vahlen matrix and it returns true or false accordingly. Any matrix with entries in a Clifford algebra is of `type/climatrix`.

A Vahlen matrix is a 2×2 matrix $V = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$ with entries in a Clifford algebra $C\ell_{p,q}$ such that the following conditions are met:

1. a, b, c, d are products of 1-vectors,
2. The pseudo-determinant[16] of V computed as $a\tilde{d} - b\tilde{c}$ equals +1 or -1;
3. $a\tilde{b}, \tilde{b}d, d\tilde{c}$, and $\tilde{c}a$ are all 1-vectors.[17]

Condition (i) above implies that $a, b, c,$ and d are elements of the Lipschitz group $L_{p,q}$ of $C\ell_{p,q}$. Recall [35] that this group is defined as follows:

$$L_{p,q} = \{ s \in C\ell_{p,q} \mid s x \hat{s}^{-1} \in \mathbb{R}^{p,q},\ x \in \mathbb{R}^{p,q} \}.$$

[16] In CLIFFORDit is computed with a prop0 d541R

Next, we consider a Vahlen matrix T that gives a translation:

```
> b:=e1+2*e3; #vector in R^(3,1)
> T:=linalg[matrix](2,2,[1,b,0,1]);
> 'isVahlenmatrix(T)'=isVahlenmatrix(T);
```

$$b := e1 + 2\ e3; \quad T := \begin{bmatrix} 1 & e1 + 2\ e3 \\ 0 & 1 \end{bmatrix}$$

$$\text{'isVahlenmatrix}(T)\text{'} = \text{true}$$

A Vahlen matrix Dil that gives a dilation transformation:

```
> delta:=1/4: #a positive parameter
> Dil:=linalg[matrix](2,2,[sqrt(delta),0,0,1/sqrt(delta)]);
> 'isVahlenmatrix(Dil)'=isVahlenmatrix(Dil);
```

$$Dil := \begin{bmatrix} \frac{1}{2} & 0 \\ 0 & 2 \end{bmatrix}$$

$$\text{'isVahlenmatrix}(Dil)\text{'} = \text{true}$$

Finally, a Vahlen matrix Tv that gives a transversion transformation:

```
> c:=2*e1-e3; #a vector in R^(3,1)
> Tv:=linalg[matrix](2,2,[1,0,c,1]);
> 'isVahlenmatrix(Tv)'=isVahlenmatrix(Tv);
```

$$c := 2\ e1 \quad e3; \quad Tv := \begin{bmatrix} 1 & 0 \\ 2\,e1 \quad e3 & 1 \end{bmatrix}$$

$$\text{'isVahlenmatrix}(Tv)\text{'} = \text{true}$$

If we now take a product of these four matrices above,[18] we will obtain an element conf of the conformal group in $R^{3,1}$:

```
> conf:=R &cm T &cm Dil &cm Tv;
```

$$conf := \begin{bmatrix} \quad \frac{e12}{2} + 10\ e23 \quad 4\,e123 & 2\,e2 \\ 2\,e123 \quad 4\,e2 & \quad 2\,e12 \end{bmatrix}$$

Since in the product above each matrix appeared exactly once, the diagonal entries of conf must be invertible. We find the inverses of each element with cinv :

```
> cinv(conf[1,1]); #inverse of conf[1,1]
```

$$\frac{2\,e12}{401} \quad \frac{40\,e23}{401}$$

[18] &cm denotes a matrix multiplication in CLIFFORD

```
> cinv(conf[2,2]); #inverse of conf[2,2]
```

$$\frac{e12}{2}$$

However, there are elements in the conformal group of $\mathbb{R}^{3,1}$ whose Vahlen matrices do not have invertible elements at all. The following example of such matrix is due to Johannes Maks. [38] Matrix W defined below represents an element in the identity component of the conformal group of $\mathbb{R}^{3,1}$:

```
> W:=evalm((1/2)*linalg[matrix](2,2,[1-e14,-e1+e4,e1+e4,1+e14]));
```

$$W := \begin{bmatrix} \frac{1}{2} - \frac{e14}{2} & -\frac{e1}{2} + \frac{e4}{2} \\ \frac{e1}{2} + \frac{e4}{2} & \frac{1}{2} + \frac{e14}{2} \end{bmatrix}$$

Notice that the diagonal elements of W are non-trivial idempotents in $C\ell_{3,1}$ hence as such they are not invertible:

```
> type(W[1,1],idempotent); #element (1,1) of W is an idempotent
```

$$true$$

```
> type(W[2,2],idempotent); #element (2,2) of W is an idempotent
```

$$true$$

Notice also that the off-diagonal elements of W are isotropic vectors in $\mathbb{R}^{3,1}$; hence they are also non-invertible. In $C\ell_{3,1}$ such vectors have zero squares:

```
> cmul(W[1,2],W[1,2]),cmul(W[2,1],W[2,1]);
```

$$0, 0$$

Let's now verify that matrix W defined above is a Vahlen matrix:

```
> 'isVahlenmatrix(W)'=isVahlenmatrix(W);
```

$$true$$

However, matrix W represents an element of the identity component of the conformal group in $\mathbb{R}^{3,1}$ since its pseudo-determinant is 1, and since it can be written as a product of a transversion, a translation, and a transversion. Thus, in another words, W is not a product of just one rotation, one translation, one dilation, and/or one transversion:

```
> Tv:=linalg[matrix](2,2,[1,0,(e1+e4)/2,1]);
```

$$Tv := \begin{bmatrix} 1 & 0 \\ \frac{e4}{2} + \frac{e1}{2} & 1 \end{bmatrix}$$

```
> T:=linalg[matrix](2,2,[1,(-e1+e4)/2,0,1]);
```

$$T := \begin{bmatrix} 1 & \frac{e4}{2} - \frac{e1}{2} \\ 0 & 1 \end{bmatrix}$$

```
> Tv &cm T &cm Tv = evalm(W); # W = Tv &cm T &cm Tv
```

$$\begin{bmatrix} \frac{1}{2}-\frac{e14}{2} & \frac{e4}{2}-\frac{e1}{2} \\ \frac{e4}{2}+\frac{e1}{2} & \frac{1}{2}+\frac{e14}{2} \end{bmatrix} = \begin{bmatrix} \frac{1}{2}-\frac{e14}{2} & \frac{e4}{2}-\frac{e1}{2} \\ \frac{e4}{2}+\frac{e1}{2} & \frac{1}{2}+\frac{e14}{2} \end{bmatrix}$$

```
> pseudodet(W); #computing pseudo-determinant of W
```

$$Id$$

Thus, the above computation confirms that $W = Tv \,\&cm\, T \,\&cm\, Tv$ and that the pseudo-determinant of W is 1:

There is another variation of Johannes Maks' example of a Vahlen matrix W without any invertible entries. Matrix W represents an element in the identity component of the conformal group of $R^{3;1}$:

```
> W:=evalm((1/2)*linalg[matrix](2,2,[1-e24,-e2+e4,e2+e4,1+e24]));
```

$$W := \begin{bmatrix} \frac{1}{2}-\frac{e24}{2} & -\frac{e2}{2}+\frac{e4}{2} \\ \frac{e2}{2}+\frac{e4}{2} & \frac{1}{2}+\frac{e24}{2} \end{bmatrix}$$

Notice that the diagonal elements of W are non-trivial idempotents in $C\ell_{3;1}$; hence they are not invertible in $C\ell_{3;1}$:

```
> type(W[1,1],idempotent); #element (1,1) of W is an idempotent
> type(W[2,2],idempotent); #element (2,2) of W is an idempotent
```

$$true\,;\, true$$

Notice also that the off-diagonal elements of W are isotropic vectors in $R^{3;1}$; hence they are also non-invertible:

```
> cmul(W[1,2],W[1,2]),cmul(W[2,1],W[2,1]);
```

$$0;\ 0$$

Finally, we verify that W is a Vahlen matrix:

```
> 'isVahlenmatrix(W)'=isVahlenmatrix(W);
```

$$'isVahlenmatrix\,(W)' = true$$

However, W is an element of the identity component of the conformal group in $R^{3;1}$ since its pseudo-determinant is 1, and since it can be written as a product of a transversion, a translation, and a transversion. As before, W is not a product of just one rotation, one translation, one dilation, and/or one transversion:

```
> Tv:=linalg[matrix](2,2,[1,0,(e2+e4)/2,1]);
```

$$Tv := \begin{bmatrix} 1 & 0 \\ \frac{e4}{2}+\frac{e2}{2} & 1 \end{bmatrix}$$

```
> T:=linalg[matrix](2,2,[1,(-e2+e4)/2,0,1]);
```

```
                              T
```

$$A := \begin{bmatrix} 2 & 3 \\ 1 & 2 \end{bmatrix}$$

Since $A \in \mathrm{Mat}(2,\mathbb{R})$, we need to find (p,q) such that $C\ell_{p,q} \simeq \mathrm{Mat}(2,\mathbb{R})$. Procedure all_sigs built into CLIFFORD displays two possible choices for the signature (p,q) such that $p+q=2$, $\mathbb{K} \simeq \mathbb{R}$ and $C\ell_{p,q}$ is a simple algebra:

```
> all_sigs(2..2,real,simple);
```

$$[[1,1],[2,0]]$$

Thus, we can pick either $C\ell_{1,1}$ or $C\ell_{2,0}$. Our choice is $C\ell_{2,0}$. We define B as the 2×2 identity matrix and use CLIFFORD's procedure clidata to display information about $C\ell_{2,0}$:

```
>
```

We are in position now to compute matrices $M_1; M_2; M_3; M_4$ representing each of the four basis elements $f1; e_1; e_2; e_{12}g$ of $C`_{2;0}$ in the basis $f f_1; f_2 g$:[22]

```
> for i to nops(clibas) do M[i]:=subs(Id=1,matKrepr(clibas[i])) end do:
```

We will use a new procedure phi which realizes the isomorphism ' from Mat$(2; R)$ to $C`_{2;0}$: This way we can find the image $p = '(A)$ in $C`_{2;0}$ of any real 2 2 matrix A: Knowing image $'(M_i)$ of each matrix M_i; $i = 1; :::; 4$; in terms of some Clifford polynomial in $C`_{2;0}$

In order to find eigenvalues and eigenvectors of $A^T A$; we will use Maple's procedure eigenvects modified by our own sorting via a new procedure assignL : The latter displays a list containing two lists: one has the eigenvalues while the second has the eigenvectors[24]. In the following, we assign the eigenvalues of $A^T A$ to

Since later we will need images of V and V^T under φ in $C\ell_{2,0}$; we compute them now and store them under the variables pV and pVt respectively.

```
> pV:=phi(V,M); #finding image of V in Cl(2,0)
> pVt:=phi(t(V),M): #finding image of t(V) in Cl(2,0)
```

pV :=

$$\left(-\frac{1}{20}\%1\sqrt{5}-\frac{1}{40}\%2\sqrt{5}+\frac{1}{8}\%2\right)Id+\left(\frac{1}{20}\%1\sqrt{5}-\frac{1}{40}\%2\sqrt{5}+\frac{1}{8}\%2\right)e1$$
$$+\left(\frac{1}{20}\%2\sqrt{5}+\frac{1}{40}\%1\sqrt{5}+\frac{1}{8}\%1\right)e2+\left(-\frac{1}{20}\%2\sqrt{5}+\frac{1}{40}\%1\sqrt{5}+\frac{1}{8}\%1\right)e12$$

$\%1 := \sqrt{10-2\sqrt{5}}$

$\%2 := \sqrt{10+2\sqrt{5}}$

The fact that V is orthogonal can be easily verified in the matrix language; in $C\ell_{2,0}$ it can be done as follows:

```
> simplify(cmul(pVt,pV));
```

$$Id$$

We repeat the above steps and apply them to AA^T. In the process, we will find its eigenvectors u_1, u_2. We must make sure that $Av_i = \sigma_i u_i$ where $\sigma_i = \sqrt{\lambda_i}$, $i = 1, 2$. This will require extra checking and possibly redefining of the u's.

```
> AAT:=evalm(A &* transpose(A)); #computing AAT
```

$$AAT := \begin{bmatrix} 13 & 8 \\ 8 & 5 \end{bmatrix}$$

The image of AA^T under φ in $C\ell_{2,0}$ we denote as ppT:

```
> ppT:=phi(AAT,M); #finding image of AAT in Cl(2,0)
```

$$ppT := 9\,Id + 4\,e1 + 8\,e2$$

In this case, the minimal polynomial of ppT and the characteristic polynomial of AA^T are, of course, the same.

```
> pol2:=charpoly(AAT,lambda); #characteristic polynomial of AAT
```

```
> pU:=phi(U,M);#finding image of U in Cl(2,0)
> pUt:=phi(t(U),M):#finding image of t(U) in Cl(2,0)
```

$$\begin{aligned}pU := {} & \left(\frac{1}{20}\%1\sqrt{5} - \frac{1}{8}\%2 - \frac{1}{40}\%2\sqrt{5}\right)\mathrm{Id} + \left(\frac{1}{20}\%1\sqrt{5} + \frac{1}{8}\%2 + \frac{1}{40}\%2\sqrt{5}\right)e1\\ & + \left(\frac{1}{20}\%2\sqrt{5} + \frac{1}{8}\%1 - \frac{1}{40}\%1\sqrt{5}\right)e2 + \left(\frac{1}{20}\%2\sqrt{5} - \frac{1}{8}\%1 + \frac{1}{40}\%1\sqrt{5}\right)e12\end{aligned}$$

$$\%1 := \sqrt{10 + 2\sqrt{5}}$$

$$\%2 := \sqrt{10 - 2\sqrt{5}}$$

The fact that U is an orthogonal matrix can be easily now checked both in the matrix language and in the Clifford language:

```
> radsimplify(evalm(t(U) &* U));#U is an orthogonal matrix
```

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

```
> simplify(pUt &c pU);
```

$$\mathrm{Id}$$

Finally, we define matrix Σ using a procedure makediag. Recall from [42] that Σ has the same dimensions as the original matrix A and that $\Sigma^T\Sigma$, $\Sigma\Sigma^T$ are the diagonal forms of A^TA and AA^T respectively. In this example matrices $\Sigma^T\Sigma$ and $\Sigma\Sigma^T$ are the same since Σ is a square diagonal matrix. Normally these matrices are different although their nonzero "diagonal" entries are the same. Therefore we have

$$\begin{gathered} A^TA = V\Sigma^T\Sigma V^T; \quad AA^T = U\Sigma\Sigma^T U^T; \quad \Sigma = \begin{bmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \end{bmatrix}; \\ \Sigma^T\Sigma = \Sigma\Sigma^T = \begin{bmatrix} \sigma_1^2 & 0 \\ 0 & \sigma_2^2 \end{bmatrix}. \end{gathered} \tag{21}$$

Matrices Σ, $\Sigma^T\Sigma$ and $\Sigma\Sigma^T$ we assign to Maple variables Sigma, STS and SST respectively:

```
> Sigma:=makediag(m,n,[seq(sigma.i,i=1..N)]);
> STS,SST:=evalm(t(Sigma) &* Sigma),evalm(Sigma &* t(Sigma));
```

$$\Sigma := \begin{bmatrix} \sqrt{5}+2 & 0 \\ 0 & \sqrt{5}-2 \end{bmatrix}$$

$$STS, SST := \begin{bmatrix} (\sqrt{5}+2)^2 & 0 \\ 0 & (\sqrt{5}-2)^2 \end{bmatrix}, \begin{bmatrix} (\sqrt{5}+2)^2 & 0 \\ 0 & (\sqrt{5}-2)^2 \end{bmatrix}$$

The corresponding images $\varphi(\Sigma)$, $\varphi(\Sigma^T\Sigma)$ and $\varphi(\Sigma\Sigma^T)$ and $C\ell_{2,0}$ will be assigned to the Maple variables pSigma, pSTS and pSST respectively:

```
> pSigma,pSTS,pSST:=phi(Sigma,M),phi(STS,M,FBgens),phi(SST,M);
```

$$pSigma, pSTS, pSST := \sqrt{5}\,\mathrm{Id} + 2\,e1,\ 9\,\mathrm{Id} + 4\sqrt{5}\,e1,\ 9\,\mathrm{Id} + 4\sqrt{5}\,e1$$

We should be able to verify in $C\ell_{2,0}$ the following two factorizations of AA^T and A^TA:

$$\begin{aligned} A^T A &= V \quad {}^T \quad V^T && (22)\\ AA^T &= U \quad {}^T U^T && (23) \end{aligned}$$

like this:

```
>
```

```
> sol:=remove(has,map(allvalues,clisolve(eigeneq,[lam          bda,x1,x2])),
> lambda=lambda);
```

sol

```
x <- cat(e,L[-1])
p1 <- substring(a1,1..(3*N-4))
p2 <- procname(x,a2,B)
S <- clibilinear(p1,p2, procname,B)
   -add((-1)^(i)*coB*nameB[L[-i],L[-1]]*
      procname(makeclibasmon(subs(L[-i]=NULL,L[1..-2])),a2,B),i=2 ..N)
return reorder(simplify(S))
end cmulNUM
```

Appendix B. Appendix: Code of cmulRS

Here is a pseudocode of the procedure cmulRS based on the combinatorial process of Rota-Stein:

```
cmulRS(x,y,B) [x, y two Grassmann monomials, B - bilinear form]
begin
 lstx <- list of indices from x
 lsty <- list of indices from y
 NX <- length of lstx
 NY <- length of lsty
 funx <- function maps integers 1..NX onto elements of lstx keeping their order
 funy <- function maps integers 1..NY onto elements of lsty keeping their order
  (this is to calculate with arbitrary indices and to compute necessary signs)
 psetx <- power set of 1..NX (actually a list in a certain order)
  (the i-th and (2^NX+1-i)-th element are disjoint adding up to the set f 1..NX g)
 psety <- power set of 1..NY (actually a list in a certain order)
  (the i-th and (2^NY+1-i)-th element are disjoint adding up to the set f 1..NY g)
  (for faster computation we sort this power sets by grade)
  (we compute the sign for any term in the power set)
 psetx <- sort psetx by grade
 psety <- sort psety by grade
 pSgnx <- sum_(i in psetx) (-1)^sum_(j in psetx[i]) (psetx[i][j]-j)
 pSgny <- sum_(i in psety) (-1)^sum_(j in psety[i]) (psety[i][j]-j)
 (we need a subroutine for cup tangle computing the bilinear form cup(x,y,B))
  begin cup
   if |x| <> |y| then return 0 end if
   if |x| = 0 then return 1 end if
   if |x| = 1 then return B[x[1],y[1]] end if
   return sum_(j in 1.. |x| )(-1)^(j1.81178(j)-1)^)8.88247(a)-1.81342(x)7.10009(,)-8.94292(1)5.31608(.)-8.94292(,)-8.94292(y)7.10009([
 pos2 <- 0
for i from 0 to min(N2,max_grade-j)
(iterate over all i-vectors of psety not exceeding max_grade while others are zero)
begin
 F2 <- N2!/((N2-i)!*i!)       (number of terms (N2 over i))
for n from 1 to F1 (for all j-vectors)
  begin
    for m from 1 to F2 (for all i-vectors)
      begin
      res <- res + pSgnx[pos1+n]*pSgny[pos2+m]*
```

References

[1] R. Ablamowicz, Clifford algebra computations with Maple . In Clifford (Geometric) Algebras with Applications in Physics, Mathematics, and Engineering, W. E. Baylis, ed. (Birkhäuser, Boston, 1996) 463{502

[2] R. Ablamowicz, Matrix exponential via Clifford algebras . J. of Nonlinear Math. Phys., 5, 3 (1998) 294{313

[3] R. Ablamowicz, Spinor Representations of Clifford Algebras: A Symbolic Approach. CPC Thematic Issue - Computer Algebra in Physics Research, Phys. Comm. 115 (1998) 510{535

[4] R. Ablamowicz, Helmstetter formula and rigid motions with CLIFFORD. In Advances in Geometric Algebra with Applications in Science and Engineering { Automatic Theorem proving, Computer Vision, Quantum and Neural Computing, and Robotics, E. Bayro-Corrochano and G. Sobczyk, eds., (Birkhäuser, Boston, 2001) 512{534

[5] R. Ablamowicz, J. Anderson and M. Baswell, Clifford algebra space singularities of inline planar platforms . In Applications of Geometric Algebra in Computer Science and Engineering, L. Dorst, ed., Chapter 36 (Birkhäuser, Boston, 2002) 463{502

[6] R. Ablamowicz and B. Fauser, On the Decomposition of Clifford Algebras of Arbitrary Bilinear Form . In Clifford Algebras and their Applications in Mathematical Physics, R. Ablamowicz and B. Fauser, eds. (Birkhäuser, Boston, 2000) 341{366

[7] R. Ablamowicz and B. Fauser, Hecke algebra representations in ideals generated by q-Young Clifford idempotents . In Clifford Algebras and their Applications in Mathematical Physics, R. Ablamowicz and B. Fauser, eds., Vol. 1: Algebra and Physics, (Birkhäuser, Boston, 2000) 245{268

[8] R. Ablamowicz and B. Fauser, CLIFFORD and BIGEBRA for Maple, http://math.tntech.edu/rafal/ (2009)

[9] R. Ablamowicz and B. Fauser, Mathematics of CLIFFORD - A Maple Package for Clifford and Grassmann Algebras . Advances in Applied Clifford Algebras, Vol. 15, No. 2 (2005), 157{181

[10] R. Ablamowicz and B. Fauser, Clifford and Grassmann Hopf algebras via the BIGEBRA package for Maple. Computer Physics Communications 170 (2005) 115{130

[11] R. Ablamowicz, B. Fauser, K. Podlaski and J. Rembielinski, Idempotents of Clifford algebras Czech. J. Phys. 53, 11 (2003) 949{955

[12] R. Ablamowicz and P. Lounesto, On Clifford algebras of a bilinear form with an antisymmetric part . In Clifford Algebras with Numeric and Symbolic Computations, R. Ablamowicz, P. Lounesto, and J. Parra, eds. (Birkhäuse

[38] J. Maks, Modulo (1,1) periodicity of Clifford algebras and the generalized (anti-) Möbius transformations . Thesis, Technische Universiteit Delft, 1989

[39] I. Porteous, Topological Geometry (Van Nostrand-Reinhold, London, 1969)

[40] G.-C. Rota and J.A. Stein, Plethystic Hopf algebras. Proc. Natl. Acad. Sci. USA 91 (1994) 13057{13061

[41] J.M. Selig, Geometrical Methods in Robotics (Springer Verlag, New York, 1996)

[42] G. Strang, Introduction to Linear Algebra (Wellesley-Cambridge Press, Wellesley, 1998)

[43] S. Tunney, On the Continuous Families of Idempotents in the Clifford Algebra of the Euclidean Plane. M.S. Thesis, Tennessee Technological University, 2003

[44] Waterloo Maple Incorporated, Maple, a general purpose computer algebra system. Waterloo, http://www.maplesoft.com (2009)

[45] F. Wright, Computing with Maple (Chapman & Hall/CRC, Boca Raton, 2002)

Rafal Ablamowicz
Department of Mathematics, Box 5054
Tennessee Technological University
Cookeville, TN 38505, USA
e-mail: rablamowicz@tntech.edu