The Maple Cliffordlib Package PhL 8.30.15

Question: Does Maple know how to do cliff things?

1. LOCATE AND INSTALL PACKAGE 1
2. PLAY WITH PACKAGE 5
3. Korman notes 8
4. Ablamowicz notes 9
5. Wiki notes on Algebra: K-algebra, subalgebra, ideal 11
6. Wheeler Notes 12

1. LOCATE AND INSTALL PACKAGE

It is not built in, but there is a package that works for Maple V located here:

http://math.tntech.edu/rafal/cliff5/index.html

Now I somewhere have notes for adding a library to Maple. As expected, I don't see such notes anywhere in my Maple folder. The notes are probably off in some folder where I actually did the job. I will now search all of Work for "Maple package". Nothing. Now search just on "package", get 355 hits. Let's try witzend on this. Try Maple AND package AND library. Get 35 hits. Try "Maple library". Nada.

But I now see that the above site has installation instructions! Here they are:

A. For Windows:

The following installation instructions should be general enough to allow for loading and installation of the package under any (hopefully) operating system. If you encounter any problems, please let me know.

- Make sure you run Maple V R. 5.1 dated November 5, 1998 [these packages should run under Maple V R. 5.0 but they have not been tested under this version of Maple. Note: This version won't run under Maple 6 or 7.].

- Download maple.ini file into the \LIB subdirectory in your Maple V directory.

[I created my own file with the single line they way in there and then put it into the Maple lib folder]
[I edit the line so it now reads: libname:="C:\Program Files\Maple V Release 5\Cliffordlib",libname:]

- Create a directory where you will store CLIFFORD's library files and help pages. Default directory name used in maple.ini file is \Cliffordlib in the directory C:\Maplev51. Thus, to use the default names, create first a directory \Maplev51 on the C:\ drive and then a subdirectory \Cliffordlib in the directory C:\Maplev51.

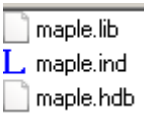
[I did what they said, my folder Cliffordlib is at top level within the Maple directory.]

• If you want to install CLIFFORD in a different directory than C:\Maplev51\Cliffordlib, you may do so but you will need to change accordingly the single line in maple.ini. Use the current line shown above as an example. Make sure you type the path correctly and appropriately for your operating system.
[did it]

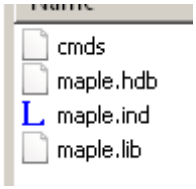
• Download file Cliffordlib.zip (just left click on the link and then save to your system) to C:\Maplev51\Cliffordliband unzip it there using, for example, on a Windows machine WinZip 8.1 (32-bit) for Windows Me/NT/2000/XP. If you don't have any unzipper, download file Cliffordlib.exe (again, just left click on the link and then save to your system) to C:\Maplev51\Cliffordlib and then "install" it, that is, run it to let it unzip automatically in the directory C:\Maplev51\Cliffordlib (make sure that you type the path C:\Maplev51\Cliffordlib into the window 'Unzip to folder:').
[I have the three files in this new folder]

• If you have any problems with installation under Windows, please contact Dr. Rafal Ablamowicz at rablamowicz@tntech.edu.

So I have to search the web afresh on this subject due to lost notes. There is no help in the little download of three files. Here are the three files I get



My current Maple lib directory has only these files



Where is my maple.ini file? I have no such file on drive C on Black.

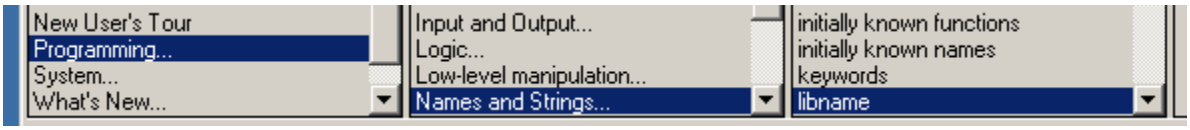
One sit says to just put the new files in a folder somewhere, then make a maple.ini which points there and put this ini into the maple lib.

Idea: Let just copy the three new files to my Maple lib area and see what happens. But ouch, they have the same file names, so cannot do it that way!

So lets fire up Maple and see what happens. At least it starts up, but then not so good:

```
> restart;
> with(Cliffordlib);
Error, (in with) undefined package , Cliffordlib
```

So the instructions were not got enough. How can I view the Maple variable libname? There is a discussion here



There is says you can view libname just by typing it, and I get

```
libname;
                    "C:Program FilesMaple V Release 5Cliffordlib", "C:\\Program Files\\Maple V Release 5/lib"
```

So I have a slashes problem here! I put this into my ini file

libname:="C:\Program Files\Maple V Release 5\Cliffordlib",libname:

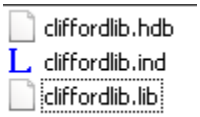
thinking this was the windows standard. Let's try this instead

libname:="C:\\Program Files\\Maple V Release 5/Cliffordlib",libname:

where I just mimic the system syntax. Well, the libname seems OK, but it cannot find it

```
> libname;
                    "C:\\Program Files\\Maple V Release 5/Cliffordlib", "C:\\Program Files\\Maple V Release 5/lib"
> with(Cliffordlib);
Error, (in with) undefined package , Cliffordlib
```

Let's try renaming all the little files replacing maple with Cliffordlib. So my little directory now has



Does this help? No, same problem. Try lower case everywhere. Still no go!

```
> libname;
                    "C:\\Program Files\\Maple V Release 5/cliffordlib", "C:\\Program Files\\Maple V Release 5/lib"
> with(cliffordlib);
Error, (in with) undefined package , cliffordlib
```

So that is not the problem.

OK, lets fire up Alta and see if I didn't add a package there! I see some dirac package files right in the lib directory, but with(dirac) does nothing, nor does readlib(dirac). Another difference is that Alta has stuff in the update folder, whereas block does not. The site above warns to get updated to Maple 5.1 November 5, 1998. Can I download an update? Both Alta and Black are Maple 5.0.

Well, I see no available Maple V updates on the web, nor at waterloo. I think I have to be happy with my version as is. What is on my Maple CD? The usual stuff, no separate update or SP items.

Now, when I look in side the files, I see Cliff5 sometimes. And then in his help thing I see this

```
> restart:with(Cliff5);
```

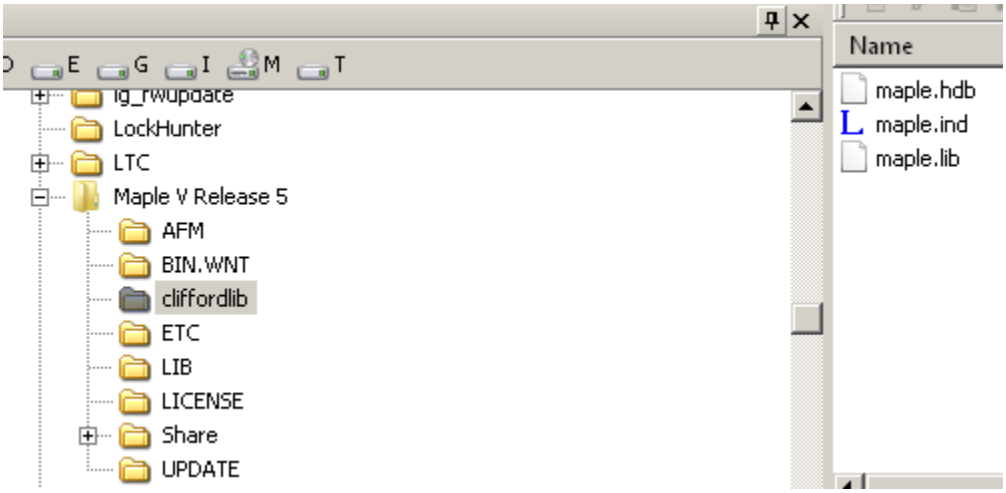
So duh, let's try that:

```
> restart;
> libname;
                    "C:\\Program Files\\Maple V Release 5/Cliffordlib", "C:\\Program Files\\Maple V Release 5/lib"
> with(Cliff5);
[&c, &cQ, &cQm, &cm, &om, &q, &qm, &rm, &w, &wm, Bsignature, CLIFFORD_ENV, Kfield, LC, LCQ, RC, RCQ, RHnumber,
   adfmatrix, all_sigs, beta_minus, beta_plus, buildm, bygrade, c_conjug, cbasis, cdfmatrix, cexp, cexpQ, cinv, clibilinear, clicollect, clidata,
   clilinear, climinpoly, cliparse, cliremove, clisolve, clisort, cliterms, cmul, cmulNUM, cmulQ, cmulRS, cmulgen, cocycle, commutingelements,
   conjugation, convert/dfmatrix, convert/mlist, convert/str_to_int, ddfmatrix, diagonalize, displayid, extract, factoridempotent, find1str,
   findbasis, gradeinv, init, isVahlenmatrix, isproduct, makealiases, makeclibasmon, matKrepr, maxgrade, maxindex, mdfmatrix,
   minimalideal, ord, permsign, pseudodet, q_conjug, qdisplay, qinv, qmul, qnorm, reorder, reversion, rmulm, rot3d, scalarpart, sexp,
   specify_constants, spinorKbasis, spinorKrepr, squaremodf, subs_clipolynom, type/antisymmatrix, type/clibasmon, type/climatrix,
   type/climon, type/clipolynom, type/cliprod, type/cliscalar, type/dfmatrix, type/diagmatrix, type/evenelement, type/fieldelement,
   type/gencomplex, type/genquatbasis, type/genquaternion, type/idempotent, type/nilpotent, type/oddelement, type/primitiveidemp,
   type/purequatbasis, type/quaternion, type/symmatrix, type/tensorprod, useproduct, vectorpart, version, wedge, wexp ]
>
```

Now we are cooking with gas! The files are back to simple maple names. Here is the ini file line which is right in the maple lib directory,

libname:="C:\\Program Files\\Maple V Release 5/Cliffordlib",libname:

and here is my little added folder



So I guess now I know how to add a package. Is the help on this working?

2. PLAY WITH PACKAGE

Note that this kind of stuff also works now

restart:with(Cliff5):with(Bigebra);

and you get that package too. The man pages are all good!

OK, here is a useful help item:

```
> restart:with(Cliff5):
> cbasis(3);
                    [Id, e1, e2, e3, e1we2, e1we3, e2we3, e1we2we3]
> cbasis(3,2);
                          [e1we2, e1we3, e2we3]
> cbasis(3,'even');
                          [Id, e1we2, e1we3, e2we3]
> cbasis(0);
                                [Id]
```

So $e_1 \wedge e_2$ is written e1we2. Makes complete sense. Identify is Id. So here you see the basis for $CA[R^3]$. Now how do you do a wedge product?

```
wedge(1+e3we2-3*ejweiwe3,2*Id+e1we2we3+3*ejweiwe3);
```

$$2\,Id - 2\,e2we3 + 3\,e3weiwej + e1we2we3$$

This is saying:

A = 1 + e_3 ^ e_2 - 3 e_j ^ e_i^e_3
B = 2 + e_1 ^ e_2 ^ e_3 + 3 e_j ^ e_i ^ e_3

The claimed result is

A^B = 2 - 2e_2^e_3+ 3e_3^e_i^e_j + e_1^e_2^ e_3

They allow undetermined integers like i and j here, velly intellesting.

Let's try the geometric product I did by hand:

A = 2 + 3e_1 + 4 e_{23} + 5 e_{123}
B = 1 + 2e_1 + 2 e_{13} + e_{123}

and after much fiddling I found *by hand* that,

AB = 3 + 3e_1 + 10e_2 + 6e_3 + 8e_{12} + 4 e_{13} + 17 e_{23} + 15 e_{123} .

Let's give it a whirl. But how do you do a geometric product in Cliff5. Here is one obscure way:

```
A := 2 + 3*e1 + 4*e2we3 + 5*e1we2we3;
```

$$A := 2 + 3\,e1 + 4\,e2we3 + 5\,e1we2we3$$

```
B := 1 + 2*e1 + 2*e1we3 + e1we2we3;
```

$$B := 1 + 2\,e1 + 2\,e1we3 + e1we2we3$$

```
K:=linalg[diag](1$4):
&c[''K''](A,B);
```

$$3\,Id + 3\,e1 + 17\,e2we3 + 15\,e1we2we3 + 4\,e1we3 + 6\,e3 + 8\,e1we2 + 10\,e2$$

Their answer (if I did it right in Maple) is thus,

3 + 3e_1 + 17 e_2e_3 + 15 $e_1e_2e_3$ + 4e_1e_3 + 6 e_3 + 8 e_1e_2 + 10e_2

which I manually reorder to get

3 + 3e_1 + 10e_2 + 6 e_3 + 8 e_1e_2 + 4e_1e_3 + 17 e_2e_3 + 15 $e_1e_2e_3$

I put in red the terms that agree with my manual result. So we agree exactly!

Perhaps the K thing is the metric tensor?

The things like e1wee2 are called "the Grassman basis monomials" by Maple.

The name of the algebra is called Cl(V,B) where V is a read vector space and B is the "bilinear form" for that vector space. I am not sure what that means. Is it a scalar product?

V is set to dim = 9 by default, I guess just to give some headroom.

Clifford product in Cl(V,B) is given by the procedure Cliff5[cmul], while the wedge product is given by the procedure Cliff5[wedge].

Let's try one wedge product and then call it quits. I used the same multivectors as above

$A = 2 + 3e_1 + 4\, e_{23} + 5\, e_{123}$

$B = 1 + 2e_1 + 2\, e_{13} + \ e_{123}$

and manually this is what I got

$A \wedge B = 2 + 7e_1 + 4\, e_{13} + 4\, e_{23} + \ 15\, e_{123}$.

Here is the Maple version (same result by two syntaxes)

```
> wedge(A,B);
```

$$2\,Id + 7\,e1 + 4\,e2we3 + 15\,e1we2we3 + 4\,e1we3$$

```
> A &w B;
```

$$2\,Id + 7\,e1 + 4\,e2we3 + 15\,e1we2we3 + 4\,e1we3$$

and I translate their result to

$2 + 7\, e_1 + 4e_1e_3 + 4\, e_2e_3 \ + 15\, e_1e_2e_3$

$= 2 + 7\, e_1 + 4e_{13} + 4\, e_{23} \ + 15\, e_{123}$

and they agree. So I got Maple to compute the two examples in my cliff review 1 doc, and things agree.

Question: What does Maple mean by saying B is "the bilinear form" ? I think this is just a 2-multilinear function. I thought Hitchin was going to help me on this since he uses a bilinear form and even calls it B, but I think that was a red herring. Can I find some general doc on the Maple package that is more than just the man pages?? Start again here

http://math.tntech.edu/rafal/cliff5/index.html

"Procedure 'cmul' and its infix form '&c' give the Clifford multiplication in the Clifford algebra of an arbitrary bilinear form B "

So what does this red phrase mean? I at least find this phrase in a Korman pdf, so take a look there.

3. Korman notes

Definition 1.1. *Given a vector space V over the field $\mathbb{F}$ with a bilinear form g, its Clifford algebra, $Cl(V)$, is the free algebra on V modulo*

$$v^2 = g(v, v). \tag{1}$$

OK, this is that idea of excluding x⊗x stuff that I have never understood. What is a "free" algebra? Is the function g(a,b) here symmetric? It must be in order to obtain result (2) below

More formally, we can construct $Cl(V)$ by quotienting out from the tensor algebra, $T(V)$, the (two-sided) ideal generated by all elements of the form $v \otimes v - g(v, v)$, for $v \in V$.

I tried on this elsewhere and could not grok this ideal. But let's continue anyway

Replacing v by $v + w$ in (1) and expanding yields

$$vw + wv = 2g(v, w), \tag{2}$$

from which we see that two vectors anti-commute if and only if they are orthogonal.

So orthogonal must mean that g(v,w) = 0 where g is the "bilinear form".

We denote the vector space $\mathbb{R}^{p+q}$ with metric signature $(\underbrace{+ + \ldots +}_{p-times}\underbrace{- - \ldots -}_{q-times})$ by $\mathbb{R}^{p,q}$. This space will be the sole focus of our work. Further, we abbreviate

Comment: Notice the word "metric" here. Korman is here assuming that the metric matrix g_{ij} is diagonal with the above diagonal elements. I show elsewhere how g_{ij} and g(x,y) are related.

If $\{e_1, \ldots, e_n\}$ is an orthonormal basis for $\mathbb{R}^{p,q}$ then by (1) and (2) these elements generate $Cl_{p,q}$ with the rules:

$$e_i^2 = \begin{cases} 1 & \text{if } 1 \leq i \leq p \\ -1 & \text{if } p+1 \leq i \leq p+q \end{cases}$$

and

$$e_i e_j = -e_j e_i \text{ if } i \neq j.$$

This is in my language at least. But what happened to g? Are they saying g is diagonal? [yes] Well, our friend Korman is off and running with me lost in the dust.

4. Ablamowicz notes

Next, I am viewing a PDF of Ablamowicz which opens with

A *quadratic form* is a map $Q : V \mapsto \mathbf{R}$ with the following properties ($\alpha \in \mathbf{R}$, $\mathbf{V} \in V$)

$$\begin{aligned} &i) \quad Q(\alpha \mathbf{V}) = \alpha^2 Q(\mathbf{V}), \\ &ii) \quad 2g(\mathbf{x}, \mathbf{y}) = Q(\mathbf{x} - \mathbf{y}) - Q(\mathbf{x}) - Q(\mathbf{y}), \end{aligned} \tag{1}$$

where $g(\mathbf{x}, \mathbf{y})$ is bilinear and necessarily symmetric. $g(\mathbf{x}, \mathbf{y})$ is called *polar bilinear form* of Q. Transposition is defined as $g(\mathbf{x}, \mathbf{y})^T = g(\mathbf{y}, \mathbf{x})$. Quadratic

Pause to look at M&M page 317:

A **bilinear form** is a second degree poly in 2n variables **x** and **y** and there are several ways to write it

$A(\mathbf{x},\mathbf{y}) = x^{\mathbf{T}} A\ y$ // M&M use an over-twiddle to mark transpose instead of T.

$= \Sigma_{ij} A_{ij} x_i y_j$ // = x • A • y in some worlds

Comment: If you break A into $A_{\mathbf{s}} + A_{\mathbf{a}}$ in the usual manner, $A_{\mathbf{a}}$ makes no contribution to A(x,y), so you might as well assume that A is symmetric.

This thing is in fact linear in x and in y. so my earlier guess was pretty good. If we set **x** = **y** we get

$A(\mathbf{x},\mathbf{x}) = x^{\mathbf{T}} A\ x = \Sigma_{ij} A_{ij} x_i x_j \equiv Q(\mathbf{x})$

This thing is a **quadratic form** , obviously not linear in x. Obviously

$Q(\alpha \mathbf{x}) = \alpha^2\ Q(\mathbf{x})$

Now one could define

$2g(x,y) = Q(x-y) - Q(x) - Q(y) = \Sigma_{ij}A_{ij}(x-y)_i(x-y)_j - \Sigma_{ij}A_{ij}x_ix_j - \Sigma_{ij}A_{ij}y_iy_j$

$= - \Sigma_{ij}A_{ij}[x_iy_j + y_ix_j]$

as done in my new Ablamowicz paper. Why is this thing useful? It is a "bilinear form", that is true, and that is what I am looking for. It is symmetric. Claims one can always diagonalize g with a choice of basis.

He then blathers for a while, then we come to

Generators and relations: Physicists and most people working in Clifford analysis prefer another construction of Clifford algebras by generators and relations [17]. One chooses a set of *generators* $\mathbf{e}_i$, images of some arbitrary basis elements $\mathbf{x}_i$ of V under the usual Clifford map $\gamma : V \mapsto C\ell(V)$ in the Clifford algebra $C\ell(V)$, and asserts the validity, in the case of $\mathbf{R} = \mathsf{R}$ or C, of the normalized, 'square law':

$$\mathbf{e}_i^2 = \pm\mathbf{1}. \tag{4}$$

Using the linearity, that is polarizing this equations by $\mathbf{e}_i \mapsto \mathbf{e}_i + \mathbf{e}_j$, one obtains the usual set of relations which have to be used to 'canonify' the algebraic expressions:

$$\mathbf{e}_i\mathbf{e}_j + \mathbf{e}_j\mathbf{e}_i = 2g(\mathbf{e}_i, \mathbf{e}_j)\mathbf{1} = 2g_{ij}\mathbf{1}. \tag{5}$$

The definition of the Clifford algebra reads:

$$C\ell(g_{ij}, V) \simeq \mathbf{Alg}(\mathbf{e}_i) \bmod \mathbf{e}_i\mathbf{e}_j = 2g_{ij}\mathbf{1} - \mathbf{e}_j\mathbf{e}_i\,. \tag{6}$$

This is a confusing statement. Maybe it means an algebra of the elements $\mathbf{e}_i$ to which we simply *add the condition* that $\{e_i,e_j\} = 2g_{ij}$ [anticommutator notation] .

OK, now maybe this is something I can chew on. From above I have

$2g(\mathbf{x},\mathbf{y}) = - \Sigma_{ij}A_{ij}[x_iy_j + y_ix_j]$ $\qquad\qquad$ $\mathbf{x} = \Sigma_i x_i\mathbf{e}_i$

$2g(\mathbf{e}_n,\mathbf{e}_m) = - \Sigma_{ij}A_{ij}[(\mathbf{e}_n)_i(\mathbf{e}_m)_j + (\mathbf{e}_m)_i(\mathbf{e}_n)_j] \equiv 2\, g_{nm}$ $\qquad$ // wrong! See comment below.

But alas, this makes no sense in terms of vector notation, compared to (5) above [see comment below]. If I try installing unit vectors, then this says

$2g(\mathbf{e}_n,\mathbf{e}_m) = - \Sigma_{ij}A_{ij}[\delta_{ni}\delta_{mj} + \delta_{nn}\delta_{nj}] = -(A_{nm} + A_{mn}) = -2\, A_{nm}$ if A is symmetric

so he has lots me and I cannot go on. So A has had his two cents. The above stuff is parroted in the wiki site on Clifford.

Comments: In R^3 one writes basis vectors $\mathbf{e}_i$ with components $(\mathbf{e}_i)_j = \delta_{ij}$. One does not talk about something like $\mathbf{e}_1\mathbf{e}_2$ because that makes no sense *in the vector space* R^3. But the e_i are also basis vectors in a different vector space which is Cl(2), say, whose basis elements are 1, e_1, e_2, e_{12}. Now within *this* vector space, the combination e_1e_2 *does* make sense, and we have in fact that $e_1e_2 = e_{12}$. So maybe it would be good to not bold e_i when you think of it as a Clif, and bold it as $\mathbf{e}_i$ when an element of R^3. This is what confused me above. Some authors, however, like Ablamowicz above, choose to embolden all elements of Cl(2) as in (5) and (6) above , where **1** is the Cl(2) identity. So then you cannot tell just by looking at the symbol whether it is meant to be an element of R^n or $C\ell(n)$. You have to tell by the context. So in my confusion above, I mix these together and get a meaningless result!

5. Wiki notes on Algebra: K-algebra, subalgebra, ideal

Here is wiki on **"Algebra over a Field"**:

Let *K* be a field, and let *A* be a vector space over *K* equipped with an additional binary operation from $A \times A$ to *A*, denoted here by · (i.e. if **x** and **y** are any two elements of *A*, **x** · **y** is the *product* of **x** and **y**). Then *A* is an **algebra** over *K* if the following identities hold for any three elements **x**, **y**, and **z** of *A*, and all elements ("scalars") *a* and *b* of *K*:

- Right distributivity: $(\mathbf{x} + \mathbf{y}) \cdot \mathbf{z} = \mathbf{x} \cdot \mathbf{z} + \mathbf{y} \cdot \mathbf{z}$
- Left distributivity: $\mathbf{x} \cdot (\mathbf{y} + \mathbf{z}) = \mathbf{x} \cdot \mathbf{y} + \mathbf{x} \cdot \mathbf{z}$
- Compatibility with scalars: $(a\mathbf{x}) \cdot (b\mathbf{y}) = (ab) (\mathbf{x} \cdot \mathbf{y})$.

These three axioms are another way of saying that the binary operation is bilinear. An algebra over *K* is sometimes also called a *K-algebra*, and *K* is called the *base field* of *A*. The binary operation is often referred to as *multiplication* in *A*. The convention adopted in this article is that multiplication of elements of

So the objects **x** and **y** above are elements of vector space A. The *scalars* are elements of field K, so that is where the K comes in. As I know, R and C are fields, as are the Galois finite fields, so I guess any of these could be K (aka the Base Field). Now why is the above called "bilinear"? For a function that means

$$f(a\mathbf{x} + b\mathbf{y}, ...) = af(\mathbf{x}, ...) + bf(\mathbf{y}, ...)$$

Let's try this for the above algebra:

$$(\alpha\mathbf{x}+\beta\mathbf{y})\bullet\mathbf{z} = \alpha\, \mathbf{x}\bullet\mathbf{z} + \beta\, \mathbf{y}\bullet\mathbf{z}$$

This what you would mean by "the operation • is linear in the first vector". But this is really the same as saying the • is "right distributive" since **z** is on the right. So if the operation • is both left and right distributive, then I agree that • is linear in both arguments, and that is where "bilinear" comes from.

There follows a possibly useful discussion of **subalgebras and ideals**.

In Galois, remember that a ring has • and + and is a field *except* it may have no mult identity and may not have a mult inverse for all elements. A "ring with identity" has the g^{-1} possibly missing. I then talked there about an **ideal** within a **ring**. r•I = I•r = I. was the key idea, where r is anywhere in the ring.

Here the discussion is a little different. Galois doc does not even use the words algebra or module. So first, what is "**an algebra**"? Wiki says it is a "structure has an addition, multiplication, and a scalar multiplication ". Notice they are not calling it a "vector space" or a "set of elements".

An "algebra over a field" is a "vector space" *equipped with* a "bilinear product" **x•y**. The vector space part means you can do vector addition and scalar multiplication, so the new feature is multiplication of vectors. The scalars are in the field. If the scalars were instead elements of a ring (where there may be scalars which have no inverses and there may be no identity scalar) , then you have a vector space over a ring, and that is what a module is. There is more to the module world, but I think that is the main idea. Note that "vector space" by default implies the related scalars are in a *field*.

An algebra over a field K (known as a **K-algebra**) may or may hot have the associative property under the • operation. So perhaps (A•B)•C ≠ A•(B•C) where • is the arbitrary op (I call * in Galois). The cross product is an example of this ≠. **Unital** = **unitary** means there is a multiplicative identity.

Suppose A is a K-algebra and we want to have L be a **subalgebra**. What does that require? For any pair x and y in L and c in K you must have x•y and y•x both in L, AND cx must be in L.

An **ideal** is a subalgebra with an extra requirement:

left-ideal:	extra requirement is that z•x is in L for any z in A (z on the left)
right-ideal:	extra requirement is that x•z is in L for any z in A (z on the right)
ideal:	both the above

In Galois I write r•I = I•r = I so this is the same idea. There r was an element of ring R, whereas here r would be an element of an algebra A, that clement called z above instead of r. Same idea!

Resume 9.1.15.

6. Wheeler Notes

Wheeler has some things to say. He works a lot with the basis vectors e_i but then he brings up the subject of what happens if they are not orthogonal. He is only thinking R^2 at this point. He says,

4. Second order Clifford algebra with general metric. We turn now to study of the implications of writing, in place of (20),

$$(g_{ij}x^i x^j) = (x^i \boldsymbol{e}_i)^2 \quad : \quad \text{all } x^i \tag{71}$$

where

$$\mathcal{g} \equiv \begin{pmatrix} g_{11} & g_{12} \\ g_{21} & g_{22} \end{pmatrix}$$

is understood to be real, symmetric and non-singular ($g \equiv \det \mathcal{g} \neq 0$) but we will not impose the requirement that $\mathcal{g}$ be positive-definite ($g > 0$). It has been my experience (in other, more complicated, contexts) that metric generalization

$$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \longmapsto \begin{pmatrix} g_{11} & g_{12} \\ g_{21} & g_{22} \end{pmatrix}$$

tends to complicate life at the outset, but that the added effort is worthwhile in the longrun, for it exposes important distinctions that otherwise remain invisible.

Immediately

$$\boldsymbol{e}_i\,\boldsymbol{e}_j + \boldsymbol{e}_j\,\boldsymbol{e}_i = 2g_{ij}\mathbf{I} \tag{72}$$

So this is the first author I have found who talks about "metric". To get the last line here, we have to back way up to an earlier section where he says,

2. Clifford algebra of order 2. This subject arises when we ask not—as at (1)—to factor but to *extract the formal square root* of $x^2 + y^2$. Or, as we find it now more convenient to notate the assignment, to extract the square root of

$$x^1x^1 + x^2x^2 \equiv \delta_{ij}x^ix^j \quad \text{where} \quad \|\delta_{ij}\| \equiv \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

To that end we posit the existence of objects $\mathbf{I}$, $\boldsymbol{e}_1$ and $\boldsymbol{e}_2$ such that

$$(\delta_{ij}x^ix^j)\mathbf{I} = (x^i\boldsymbol{e}_i)^2 \quad : \quad \text{all } x^i \tag{20}$$

Immediately

$$\begin{aligned} \boldsymbol{e}_i\,\boldsymbol{e}_j + \boldsymbol{e}_j\,\boldsymbol{e}_i &= (\delta_{ij} + \delta_{ji})\mathbf{I} \\ &= 2\delta_{ij}\mathbf{I} \quad \text{by} \quad \delta_{ij} = \delta_{ji} \end{aligned} \tag{21}$$

What does 20 say? I will put indices down for the moment just to make typing easier.

$\mathbf{x} = \Sigma_i\, x_i\mathbf{e}_i$ would be an expansion of a vector $\mathbf{x}$

The left side of (20) says $(x_1^2 + x_2^2)$I. So then (20) reads: $(x_1^2 + x_2^2)\mathrm{I} = \mathbf{x}\,\mathbf{x} = x^2$ which seems pretty reasonable as a thing to "posit". Now write out the RHS

$\mathbf{x}^2 = (x_1e_1 + x_2e_2)(x_1e_1 + x_2e_2) = x_1^2e_1^2 + x_2^2e_2^2 + 2(e_1e_2 + e_2e_1)$

So then we have

$(x_1^2 + x_2^2)1 = x_1^2e_1^2 + x_2^2e_2^2 + 2(e_1e_2 + e_2e_1)$

Write out more generally:

$\Sigma_i x_i^2\,\mathrm{I} = (\Sigma_i x_i e_i)(\Sigma_j x_j e_j) = \Sigma_i\,[\,\Sigma_j\, x_i e_i x_j e_j]$

Alternately,

$\Sigma_i x_i^2\,\mathrm{I} = (\Sigma_j x_j e_j)(\Sigma_i x_i e_i) = \Sigma_j\,[\,\Sigma_i\, x_j e_j x_i e_i] = \Sigma_i\,[\,\Sigma_j\, x_j e_j x_i e_i]$

Add these two equations to get

$$2\,\Sigma_i x_i^2\, I = \Sigma_i[\,\Sigma_j\, x_i x_j\,(\,e_i e_j + e_j e_i)]$$

For this to be true for any x_i components, we would want to have

$$e_i e_j + e_j e_i = 2\,\delta_{ij} I$$

and this is (21). Setting i = j gives $e_i^2 = 1$ and setting $i \neq j$ gives $e_i e_j = -\,e_j e_i$, as he goes on to say,

which when spelled out in specific detail read

$$\boldsymbol{e}_1\,\boldsymbol{e}_1 = \boldsymbol{e}_2\,\boldsymbol{e}_2 = \mathsf{I} \tag{22.1}$$

$$\boldsymbol{e}_1\,\boldsymbol{e}_2 + \boldsymbol{e}_2\,\boldsymbol{e}_1 = \mathbf{0} \tag{22.2}$$

He goes on with very clear statements:

It follows that products of the general form $\boldsymbol{e}_{i_1}\boldsymbol{e}_{i_2}\boldsymbol{e}_{i_3}\cdots\boldsymbol{e}_{i_n}$, which we suppose to have been assembled from p $\boldsymbol{e}_1$'s and $q = n-p$ $\boldsymbol{e}_2$'s, can (by $\boldsymbol{e}_2\,\boldsymbol{e}_1 = -\,\boldsymbol{e}_1\boldsymbol{e}_2$) always be brought to "dictionary order"

$$\pm \underbrace{\boldsymbol{e}_1\boldsymbol{e}_1\cdots\boldsymbol{e}_1}_{p\text{ factors}}\underbrace{\boldsymbol{e}_2\boldsymbol{e}_2\cdots\cdots\boldsymbol{e}_2}_{q\text{ factors}}$$

where $(\pm) = (-)^{\text{number of transpositions required to achieve dictionary order}}$. Drawing now upon (22.1) we find that the expression presented just above can be written

$$= \pm \begin{cases} \mathsf{I} & \text{if } p \text{ even, } q \text{ even} \\ \boldsymbol{e}_1 & \text{if } p \text{ odd, } q \text{ even} \\ \boldsymbol{e}_2 & \text{if } p \text{ even, } q \text{ odd} \\ \boldsymbol{e}_1\boldsymbol{e}_2 & \text{if } p \text{ odd, } q \text{ odd} \end{cases}$$

and that this list exhausts the possibilities. We confront therefore an algebra with elements of the form

$$\boldsymbol{a} = a^0\mathsf{I} + a^1\boldsymbol{e}_1 + a^2\boldsymbol{e}_2 + a^{12}\boldsymbol{e}_1\boldsymbol{e}_2 \tag{23}$$

So what is he saying here? In an "algebra" you will consider lots of products of elements of the algebra. When you do such products, you will end up with products of the e_i . But all these basis vector products boil down to only 4 possible forms as listed above. Thus, it must be that (23) is the most general form you can have for any element of the algebra. This I recognize as scalar + vector or Cl(2).

Now let's stop that thread and jump ahead to page 17 where we were above, and where he says (I repeat),

4. Second order Clifford algebra with general metric. We turn now to study of the implications of writing, in place of (20),

$$(g_{ij}x^i x^j) = (x^i \boldsymbol{e}_i)^2 \quad : \quad \text{all } x^i \tag{71}$$

where

$$\mathbf{g} \equiv \begin{pmatrix} g_{11} & g_{12} \\ g_{21} & g_{22} \end{pmatrix}$$

is understood to be real, symmetric and non-singular ($g \equiv \det \mathbf{g} \neq 0$) but we will not impose the requirement that $\mathbf{g}$ be positive-definite ($g > 0$). It has been

which compare to the earlier result // note he says g is symmetric!

$$(\delta_{ij}x^i x^j)\mathbf{I} = (x^i \boldsymbol{e}_i)^2 \quad : \quad \text{all } x^i \tag{20}$$

M&M would refer to $\delta_{ij}x_i x_j$ as a **quadratic form** with A = 1

$$A(\mathbf{x},\mathbf{x}) = x^T A\, x = \Sigma_{ij} A_{ij} x_i x_j \equiv Q(\mathbf{x})$$

and we are generalizing this from A = 1 to A = g. As noted above, you might as well assume g is symmetric. So g here is a "quadratic form" matrix and for me it is "similar to" a metric tensor, but we don't have $ds = g_{ij}dx_i dx_j$, we have this for full vector components, not differential ones. So perhaps you could define A(x,x) = the norm $||x||^2$ and then $|| x - y||^2$ would be the "natural metric". Then

$$Q(\mathbf{x}) = A(\mathbf{x},\mathbf{x}) = ||\mathbf{x}||^2$$

$$d(\mathbf{x},\mathbf{y})^2 = || \mathbf{x} - \mathbf{y}||^2 = Q(\mathbf{x}-\mathbf{y}) = \Sigma_{ij} A_{ij}(x_i - y_i)(x_j - y_j) = A(x,x) + A(y,y) - 2\,A(x,y)$$

$$= Q(\mathbf{x}) + Q(\mathbf{y}) - 2A(\mathbf{x},\mathbf{y})$$

where now we have defined a new object which is

$$A(\mathbf{x},\mathbf{y}) = \Sigma_{ij} A_{ij} x_i y_j = \text{"a bilinear form".}$$

Here if $A = A_s + A_a$ you could have A_a playing a role, but it seems to me we already threw out A_a.

So if we do this, then you could say that A(x,y) is *related to* the metric d(**x**,**y**) . Note that

$$Q(\mathbf{x}-\mathbf{y}) = Q(\mathbf{x}) + Q(\mathbf{y}) - 2A(\mathbf{x},\mathbf{y})$$

or

$$2A(\mathbf{x},\mathbf{y}) = Q(\mathbf{x}) + Q(\mathbf{y}) - Q(\mathbf{x}-\mathbf{y})$$

Now recall from above the **Ablam** statement that

A *quadratic form* is a map $Q : V \mapsto \mathbf{R}$ with the following properties ($\alpha \in \mathbf{R}$, $\mathbf{V} \in V$)

$$\begin{aligned} i) \quad & Q(\alpha \mathbf{V}) = \alpha^2 Q(\mathbf{V}), \\ ii) \quad & 2g(\mathbf{x},\mathbf{y}) = Q(\mathbf{x}-\mathbf{y}) - Q(\mathbf{x}) - Q(\mathbf{y}), \end{aligned} \tag{1}$$

where $g(\mathbf{x},\mathbf{y})$ is bilinear and necessarily symmetric. $g(\mathbf{x},\mathbf{y})$ is called *polar bilinear form* of Q. Transposition is defined as $g(\mathbf{x},\mathbf{y})^T = g(\mathbf{y},\mathbf{x})$. Quadratic

Conclusion: this "polar bilinear form of Q" g(x,y) is related to the natural "metric tensor" for the metric as defined above

$$d(\mathbf{x},\mathbf{y})^2 = Q(\mathbf{x}) + Q(\mathbf{y}) - 2g(\mathbf{x},\mathbf{y})$$

where d(x,y) is the metric. So g(x,y) is not itself "the metric" , it is related to the natural metric d(**x**,**y**). I don't really see any metric *tensors* here.

So back to Wheeler again:

4. Second order Clifford algebra with general metric. We turn now to study of the implications of writing, in place of (20),

$$(g_{ij}x^i x^j) = (x^i \boldsymbol{e}_i)^2 \quad : \quad \text{all } x^i \tag{71}$$

where

$$\boldsymbol{g} \equiv \begin{pmatrix} g_{11} & g_{12} \\ g_{21} & g_{22} \end{pmatrix}$$

I think we just repeat the math we did earlier and we arrive at this point (adding two equations)

$$2\,\Sigma_i g_{ij} x_i x_j\, \mathrm{I} = \Sigma_i [\,\Sigma_j\, x_i x_j\, (\, e_i e_j + e_j e_i)]$$

For this to be true for any x_i components, we would want to have

$$e_i e_j + e_j e_i = 2\, g_{ij} \mathrm{I}$$

and now we no longer have simple alternation negation of $e_i e_j$. So the suggestion is made to define a NEW object e_{ij} as follows:

Immediately

$$\boldsymbol{e}_i \boldsymbol{e}_j + \boldsymbol{e}_j \boldsymbol{e}_i = 2g_{ij}\mathbf{I} \tag{72}$$

The antisymmetry condition (22.2) is now lost, though we have in its place the antisymmetry of

$$\boldsymbol{e}_{ij} \equiv \boldsymbol{e}_i \boldsymbol{e}_j - g_{ij}\mathbf{I} \quad : \quad \boldsymbol{e}_{ij} = -\boldsymbol{e}_{ji} \tag{73}$$

I verify:

$e_{ij} = e_i e_j - g_{ij}$

$e_{ji} = e_j e_i - g_{ji} = [\, 2g_{ij} - e_i e_j] - g_{ji} = - [e_i e_j - g_{ij}] = - e_{ij}$ of g is symmetric!

He now writes

$f = (1/2)\, \varepsilon^{ij} e_{ij} = (1/2)\, [\, e_{12} - e_{21}] = e_{12}$ with this new e^{ij}

and he then continues:

It becomes therefore natural to close the $\{e_i, e_j\}$-generated algebra with the introduction of

$$\begin{aligned} f &\equiv \tfrac{1}{2}\varepsilon^{ij} e_{ij} && (74.1)\\ &= e_{12} \\ &= \quad e_1 e_2 - g_{12} \mathbf{I} && (74.2)\\ &= -e_2 e_1 + g_{21} \mathbf{I} && (74.3) \end{aligned}$$

From (72)–(74) we extract the primative products

$$\begin{aligned} e_1 e_1 &= g_{11}\mathbf{I} & e_1 e_2 &= g_{12}\mathbf{I} + f & e_1 f &= g_{11} e_2 - g_{12} e_1 \\ e_2 e_1 &= g_{21}\mathbf{I} - f & e_2 e_2 &= g_{22}\mathbf{I} & e_2 f &= g_{21} e_2 - g_{22} e_1 \\ f\, e_1 &= g_{21} e_1 - g_{11} e_2 & f\, e_2 &= g_{22} e_1 - g_{12} e_2 & f\, f &= -g\mathbf{I} \end{aligned}$$

where in the final equation $g \equiv g_{11}g_{22} - g_{12}g_{21} = \det \mathbf{g}$. Therefore and equivalently:

Now he is just "working out" the multiplication table for this little CL(2) algebra. In the last item

$f\, f = e_{12} e_{12} = (e_1 e_2 - g_{12})\,(e_1 e_2 - g_{12}) = e_1 e_2 e_1 e_2 - g_{12}\, e_1 e_2 - g_{12}\, e_1 e_2 + g_{12}^2$

$= e_1 e_2 e_1 e_2 - 2\, g_{12}\, e_1 e_2 + g_{12}^2$

But now write

$e_2 e_1 = - e_1 e_2 + 2 g_{12}$

Then we get

$ff = e_1 e_2 e_1 e_2 - 2\, g_{12}\, e_1 e_2 + g_{12}^2$

$= e_1 [- e_1 e_2 + 2 g_{12}] e_2 - 2\, g_{12}\, e_1 e_2 + g_{12}^2$

$= - g_{11} g_{22} + 2 g_{12} e_1 e_2 - 2\, g_{12}\, e_1 e_2 + g_{12}^2$

$= - g_{11}g_{22} + g_{12}^2 = - [g_{11}g_{22} - g_{12}^2] = - \det(g) \equiv - g$

He then goes on to write c and C as arbitrary elements of this Cl(2) and it is closed.

Wheeler then carries on where I don't care to follow right now. He is using the real tensor algebra notation with up and down indices, he talks about ε having weight 1 and g having weight 2 and all that stuff that is no mystery to me anymore. But right now I am just interested in his metric related matrix g. In his world it is just a 2x2 matrix. But what I have seem above is that this matrix defines a bilinear form in this manner [recall isomorphism comment of an earlier author] .

$g(\mathbf{x},\mathbf{y}) = \Sigma_{ij} g_{ij} x_i y_j$

so one could say that Wheeler is talking about "Cℓ(2) of an arbitrary bilinear form g(x,y) "

I *suspect* that this generalizes to the following:

Cℓ(R^n; g) g = bilinear form $\boldsymbol{e}_i \boldsymbol{e}_j + \boldsymbol{e}_j \boldsymbol{e}_i = 2 g_{ij} \mathbf{I}$

$g(\mathbf{x},\mathbf{y}) = \Sigma_{ij} g_{ij} x_i y_j$ = bilinear form

but no one has said this so far. I would then expect some complicated definitions for e_{ij} and e_{ijk} and so on, with some very messy multiplication table. But the e_{ij} piece would probably be the same as above.

Comment: Suppose we were to write $u = \Sigma_i u_i e_i$ and $v = \Sigma_k v_k e_k$. Then

$uv+vu = (\Sigma_i u_i e_i)(\Sigma_k v_k e_k) + (\Sigma_k v_k e_k)(\Sigma_i u_i e_i)$

$= \Sigma_{ik}\, u_i v_k [e_i e_k + e_k e_i] = \Sigma_{ik}\, u_i v_k\, 2 g_{ik} = 2 \Sigma_{ik}\, g_{ik} u_i v_k = 2\, g(u,v)$

Therefore we have

$e_i e_j + e_j e_i = 2\, g_{ij} \quad \Rightarrow \quad uv + vu = 2\, g(u,v) \quad \Leftrightarrow \quad v^2 = g(v,v)$

Now what happens to wedge products if we use this general g(x,y) thing? Wheeler has no wedges until page 29 after doing rotation and Lorentz group connections.

Wheeler says that the **wedge product** should always have this definition,

$$\boldsymbol{e}_{i_1 i_2 \ldots i_p} \equiv \tfrac{1}{p!}\, \boldsymbol{e}_{i_1} \wedge \boldsymbol{e}_{i_2} \wedge \cdots \wedge \boldsymbol{e}_{i_p}$$
$$\equiv \tfrac{1}{p!} \left\{ \text{antisymmetrized} \sum_{\text{permutations}} \right\} \qquad (99)$$

For example, he wants

$\varepsilon_{123} \equiv (1/6)\ [\ e_1 \wedge e_2 \wedge e_3]\ =\ (1/6)\ [\ e_1e_2e_3 + \text{signed permutations}\ ]$

$=\ (1/6)\ [\ e_1e_2e_3 - e_2e_1e_3 + ...]$

So his definition of the wedge product then has nothing directly to do with g_{ij}. However, the g_{ij} metric thing is injected into the theory in the expected way

$$(g_{ij}x^ix^j) = (x^i\boldsymbol{e}_i)^2 \quad \text{whence} \quad \boldsymbol{e}_i\boldsymbol{e}_j + \boldsymbol{e}_j\boldsymbol{e}_i = 2g_{ij}\mathbf{I}$$

so when you try to combine terms in the antisym wedge product, various g_{ij} things will appear. So he is really saying three things above:

1. The connection that $e_{ijk} \equiv\ e_i \wedge e_j \wedge e_k$ which we used earlier remains as it was.

2. The wedge product is defined as it has always been defined as the sum shown.

3. The connection $e_{ijk} \equiv\ e_ie_je_k$ is now no longer true, so *this* is what is different.

He shows that this approach replicates his n = 2 Clifford algebra theory presented earlier.

For the n = 2 case he finds

$$\begin{aligned}\boldsymbol{e}_{123} &= \tfrac{1}{3!}(\boldsymbol{e}_1\boldsymbol{e}_2\boldsymbol{e}_3 - \boldsymbol{e}_1\boldsymbol{e}_3\boldsymbol{e}_2 + \boldsymbol{e}_2\boldsymbol{e}_3\boldsymbol{e}_1 - \boldsymbol{e}_2\boldsymbol{e}_1\boldsymbol{e}_3 + \boldsymbol{e}_3\boldsymbol{e}_1\boldsymbol{e}_2 - \boldsymbol{e}_3\boldsymbol{e}_2\boldsymbol{e}_1) \\ &= \tfrac{6}{3!}(\boldsymbol{e}_1\boldsymbol{e}_2\boldsymbol{e}_3 - g_{23}\boldsymbol{e}_1 + g_{31}\boldsymbol{e}_2 - g_{12}\boldsymbol{e}_3)\end{aligned}$$

so you see how e_{123} differs from $e_1e_2e_3$. He then says for an arbitrary element in his CL(3),

$$\boldsymbol{c} = c\,\mathbf{I} + c^i\boldsymbol{e}_i + \tfrac{1}{2!}c^{ij}\boldsymbol{e}_{ij} + \tfrac{1}{3!}c^{ijk}\boldsymbol{e}_{ijk} \qquad (100)$$

Wheeler seems to be cutting new ground with his proposals. This paper is 2003, fairly recent. He ends up on page 31 with this sort of proposal for the lower Clifford algebras,

$$
\begin{aligned}
\text{in } \mathcal{C}_2[\mathcal{g}] \quad &: \quad \boldsymbol{c} = s\,\mathbf{I} + s^i \boldsymbol{e}_i + p\,\boldsymbol{f} \\
\text{in } \mathcal{C}_3[\mathcal{g}] \quad &: \quad \boldsymbol{c} = s\,\mathbf{I} + s^i \boldsymbol{e}_i + p^i \boldsymbol{f}_i + p\,\boldsymbol{f} \\
\text{in } \mathcal{C}_4[\mathcal{g}] \quad &: \quad \boldsymbol{c} = s\,\mathbf{I} + s^i \boldsymbol{e}_i + \tfrac{1}{2!} s^{ij} \boldsymbol{e}_{ij} + p^i \boldsymbol{f}_i + p\,\boldsymbol{f} \\
\text{in } \mathcal{C}_5[\mathcal{g}] \quad &: \quad \boldsymbol{c} = s\,\mathbf{I} + s^i \boldsymbol{e}_i + \tfrac{1}{2!} s^{ij} \boldsymbol{e}_{ij} + \tfrac{1}{2!} p^{ij} \boldsymbol{f}_{ij} + p^i \boldsymbol{f}_i + p\,\boldsymbol{f} \\
\text{in } \mathcal{C}_6[\mathcal{g}] \quad &: \quad \boldsymbol{c} = s\,\mathbf{I} + s^i \boldsymbol{e}_i + \tfrac{1}{2!} s^{ij} \boldsymbol{e}_{ij} + \tfrac{1}{3!} s^{ijk} \boldsymbol{e}_{ijk} + \tfrac{1}{2!} p^{ij} \boldsymbol{f}_{ij} + p^i \boldsymbol{f}_i + p\,\boldsymbol{f}
\end{aligned}
$$

where he gives new names to things. We had f = e_{12} in the C_2 case, and then f_i and then f_{ij} as you build up to higher order. He then tries to carry out this plan explicitly for n = 2 then n = 3 then n = 4. He uses Mathematica by the way, starting on page 36. Here is an interesting fact:

DIGRESSION: Some *Mathematica* Technique. The properties —except for the transformation properties (weight)—that we associate with the Levi-Civita symbol $\epsilon_{i_1 i_2 \ldots i_n}$ are reproduced in *Mathematica* by the command `Signature[{i, j, ..., k}]`, the action of which is illustrated below:

$$
\begin{aligned}
\texttt{Signature}[\{1,2,3\}] &= +1 \\
\texttt{Signature}[\{1,1,3\}] &= \;\; 0 \\
\texttt{Signature}[\{2,1,3\}] &= -1
\end{aligned}
$$

The command is powerful enough to read subscripts, thus

$$
\begin{aligned}
\texttt{Signature}[\{\alpha_1,\alpha_2,\alpha_3\}] &= +1 \\
\texttt{Signature}[\{\alpha_1,\alpha_1,\alpha_3\}] &= \;\; 0 \\
\texttt{Signature}[\{\alpha_2,\alpha_1,\alpha_3\}] &= -1
\end{aligned}
$$

Does Maple have something like this? I recall rolling my own once. Well in Maple you can do this

```
restart; with(tensor):
Levi_Civita(1,3,a,b);
eps := a[compts];
```

$$eps := a_{compts}$$

```
eps[1,2,3];
```

1

```
eps[2,1,3];
```

-1

```
eps[2,1,1];
```

0

where we use the fancy call for a simple case.

So Wheeler goes on to show how he can use Mathematic to compute things for his clifford stuff.

Task: Let's now see if perhaps Maple's B matrix is the same as Wheeler's g matrix. What test can we do?

$e_1 e_2 = - e_2 e_1 + 2 g_{12}$

So I could try this

cmul(e1,e2) + cmul(e2,e1) which should be - 2 g_{12}

```
cmul(e1,e2) + cmul(e2,e1);
```

$$B_{1,2}\,Id + B_{2,1}\,Id$$

Looks good! Maple does not assume symmetry for its version of the g matrix B.

Now what would Maple say about this:

$e_1 \wedge e_2 = (1/2)\,[e_1e_2 - e_2e_1]$

I would then predict that

$e_1e_2 - e_2e_1 = 2\,e_1 \wedge e_2$

and here is what Maple says,

```
e1 &c e2  - e2 &c e1;
```

$$2\,e1we2 + B_{1,2}\,Id - B_{2,1}\,Id$$

So in general, if you ask Maple to do some wedge product, it does not expand that in any way. It regards that as the sort of basis for things. For example,

```
wedge(e1,e2,e3);
```

$$e1we2we3$$

It is whey you do geometric products that the B elements appear.

Now let's go back to the clifford Maple overview"

- This package allows the user to perform a variety of algebraic computations in the Clifford algebra Cl(V,B) on a real vector space V endowed with a bilinear form B. The dimension of V is must be between 1 and 9, while the bilinear form B may be undefined, numeric, symbolic, symmetric, or antisymmetric. In fact, in most computations the form B doesn't even have to be assigned in which case coefficients of B appear in the Maple output. To use any function from the package, load the package using the command with(Cliff5).

So now FINALLY I think I know that this paragraph is saying.

So here is how you make Maple do what you want for G = g = δ_{ij}:

```
restart; with(Cliff5):
B:=linalg[diag](1$3):
print(B);
```

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

```
e1 &c e2  - e2 &c e1;
```

$$2\,e1we2$$

```
e1 &c e2  + e2 &c e1;
```

$$0$$

So this gets rid of B_{ij} stuff and gives you results with the usual "orthogonal basis" that most of my pdf's use.

Now my Maple program is pretty simple:

```
restart; with(Cliff5):
B:=linalg[diag](1$9):     # set bilinear form to identity matrix
a := 2 + 3*e1 + 4*e2we3 + 5*e1we2we3;
```

$$a := 2 + 3\,e1 + 4\,e2we3 + 5\,e1we2we3$$

```
b := 1 + 2*e1 + 2*e1we3 + e1we2we3;
```

$$b := 1 + 2\,e1 + 2\,e1we3 + e1we2we3$$

```
a &c b;     # compute geometric product ab
```

$$3\,Id + 3\,e1 + 17\,e2we3 + 15\,e1we2we3 + 4\,e1we3 + 6\,e3 + 8\,e1we2 + 10\,e2$$

```
a &w b;     # compute wedge product a^b
```

$$2\,Id + 7\,e1 + 4\,e2we3 + 15\,e1we2we3 + 4\,e1we3$$

and these agree with my two hand calculations as noted above. So I think I can make Maple compute things I want computed for the orthogonal basis stuff.