VAGC_grad_Problem5.2.2_xa.wxm

A short development document for Geometric Algebra with wxMaxima
just to test some calculus functions within the GAwxM environment,
contains...
Initialization
Loading of functions (intrinsic and GA specific)
Pseudoscalar definition (specifies the space dimension) and
Calculation of the inverse pseudoscalar used to generate the dual of a multivector
Enumeration of the standard basis for the specified dimension

Problem 5.2.2, VAGC page 62 for the multivector gradient

Initialization

```
(%i42) ext:["wxm"]$
       file_type_maxima:append(ext,file_type_maxima)$
       batchload("initialize_fns")$
```

the pseudoscalar and its inverse
the lowest useable dimension pseudoscalar should be {e1,e2} i.e. Plen = 2
e.g. for four dimensions edit Pseudos:{e1,e2,e3}$ to Pseudos:{e1,e2,e3,e4}$

```
(%i1)  Pseudos:{e1,e2,e3}$
       Pvar:listofvars(Pseudos)$
       Plen:length(Pvar)$
       I:Pseudos$
       ni:(Plen-1)*Plen/2$
       Ii:(-1)^ni*I$
       kill(ni)$
       ldisplay(Pvar)$

 (%t8) Pvar=[e1,e2,e3]

(%i9)  batchload("initialize_lsts")$

 (%t9) lstblds=[[{e1},{e2},{e3}],[{e1,e2},{e1,e3},{e2,e3}],[{e1,e2,e3}]]
(%t10) allblds=[{e1},{e2},{e3},{e1,e2},{e1,e3},{e2,e3},{e1,e2,e3}]
(%t11) invblds=[{e1},{e2},{e3},-{e1,e2},-{e1,e3},-{e2,e3},-{e1,e2,e3}]
```

end of Initialization

set derivabbrev:false$

```
(%i12) derivabbrev:false$
```

Problem 5.2.2
VAGC page 62

form the coordinate vector, x and the constant vector, a from the lists of coefficients

```
(%i13) xstr:"x"$
       xlst:lstvector(xstr)$
       ldisplay(xlst)$

(%t15) xlst=[x1,x2,x3,0,0,0,0]

(%i16) x:makevector(xlst)$
       ldisplay(x)$

(%t17) x={e3}*x3+{e2}*x2+{e1}*x1

(%i18) astr:"a"$
       alst:lstvector(astr)$
       ldisplay(alst)$

(%t20) alst=[a1,a2,a3,0,0,0,0]

(%i21) a:makevector(alst)$
       ldisplay(a)$

(%t22) a=a3*{e3}+a2*{e2}+a1*{e1}

(%i23) eJ:allblds$
```

form the function, F(x)=xa in Problem 5.2.2a

```
(%i24) F(x):=x&*a$
       F:ev(F(x))$
       ldisplay(x,F)$

(%t26) x={e3}*x3+{e2}*x2+{e1}*x1
(%t27)/R/ F=(a3-{e2,e3}*a2-{e1,e3}*a1)*x3+({e2,e3}*a3+a2-{e1,e2}*a1)*x2+
({e1,e3}*a3+{e1,e2}*a2+a1)*x1

(%i28) Fstr:"F"$
       gradF:mvgrad(Fstr,xlst)$
       ldisplay(gradF)$

(%t30) gradF={e3}&*(d/(d*x3)*F)+{e2}&*(d/(d*x2)*F)+{e1}&*(d/(d*x1)*F)

(%i31) lhs:ev(gradF,diff);

(%o31)/R/ 3*a1*{e1}+3*a3*{e3}+3*a2*{e2}
```

confirm that the evaluated gradF is the same as the value given in the Problem

```
(%i32) n:Plen$
       rhs:n*a$
       is(equal(lhs,rhs));

(%o34) true
```

form the function, F(x)=x.a in Problem 5.2.2b but use the same gradient expression, gradF

```
(%i35) F(x):=x&.a$
       F:ev(F(x))$
       ldisplay(F)$

(%t37)/R/ F=a3*x3+a2*x2+a1*x1

(%i38) lhs:ev(gradF,diff);

(%o38)/R/ a1*{e1}+a3*{e3}+a2*{e2}

(%i39) rhs:a$
       is(equal(lhs,rhs));

(%o40) true
```

form the function, F(x)=x^a in Problem 5.2.2b cont.

```
(%i41) F(x):=x&^a$
       F:ev(F(x))$
       ldisplay(F)$

(%t43)/R/ F=(-{e2,e3}*a2-{e1,e3}*a1)*x3+({e2,e3}*a3-{e1,e2}*a1)*x2+
({e1,e3}*a3+{e1,e2}*a2)*x1

(%i44) lhs:ev(gradF,diff);

(%o44)/R/ 2*a1*{e1}+2*a3*{e3}+2*a2*{e2}

(%i45) rhs:(n-1)*a$
       is(equal(lhs,rhs));

(%o46) true
```